République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A-MIRA BEJAÏA

Faculté de Technlogie
Département de Génie électrique

Mémoire de fin de cycle

En vue de l'Obtention du Diplôme Master en Automatisme Industriel

THÈME

Automatisation et simulation en 3D avec implémentation d'une partie d'une chaîne de production - Cas SARL Ifri -

Présenté par : Hebbadj Abdenour

Devant le jury composé de :

Mr A/Malek. AZZI	Rapporteur	U. A/Mira Béjaia.
Mr Yazid. ACHOUR	President	U. A/Mira Béjaia.
Mr Ahmed. MELLAHI	Examinateur	U. A/Mira Béjaia.

Juin 2013

*R*emerciements

Je tiens en premier lieu à remercier les membres du jury qui m'ont fait l'honneur d'accepter de rapporter mon travail.

Je tiens à remercier M.Azzi, pour m'avoir encadré durant ce projet, m'avoir aidé à le mener à bien et pour m'avoir accordé son confiance tout au long de celui-ci.

Je tiens à exprimer ma reconnaissance à ma famille pour son soutien inconditionnel et plus particulièrement à mes parents pour leurs encouragements et leur confiance tout au long de mes années d'études.

C'est également pour moi une occasion de faire part de toute ma gratitude à tout le personnel de l'unité de IFRI pour sa disponibilité et son accueil chaleureux.

La concrétisation de ce travail n'aurait jamais vu le jour sans la précieuse collaboration et sans une volonté d'une dynamique équipe, mes amis (es), mes collègues et mes proches et dans le souci de n'oublier personne, que tout ceux qui nous ont aidé, de pris ou de loin, trouvent l'expression de notre profonde gratitude.

A. Hebbadj

La Science peut être considérée comme un problème de minimum :
Elle consiste à exposer des faits avec la moindre dépense intellectuelle.

E. Mach

Dédicaces

À toute ma famille, en particulier ma soeur *Hayet* ,

À tous mes ami(e)s sans exception aucune,

À tous ceux qui m'ont aidé et soutenu,

Aux étudiants de l'automatisme industriel,

Pour la patience, le soutien et la comprehension qu'elle m'a témoigné tout au long de ce travail ma chère *Siham* .

Je dédie ce travail.

A. Hebbadj

Table des matières

Liste des tableaux

Table des figures

Introduction Générale

Pendant les dernières décennies, grâce au progrès technologique lié principalement à la rapidité de traitement des données et les grandes capacités de stockage de l'information, les industries, toutes catégories confondues, ont considérablement évolués grâce aux moyens de haute technologie appliquée aux domaines de la gestion de production, de supervision et bien d'autres. L'usage de l'automatisation et l'informatisation ont profondément modifié le cadre de la production qui tend à remplacer l'homme dans toutes ses tâches, d'une part, et d'autre part, la forte compétitivité et la globalisation du marché ont conduit à la réalisation des systèmes de production plus hétérogènes et plus complexes qu'auparavant.

Tout système de production est composé d'une ***partie opérative*** dans laquelle l'ensemble des opérations sont réalisées et la ***partie commande*** qui donne les ordres d'exécution à la partie opérative. Dès qu'une tâche est effectuée, celle-ci informe la partie commande de sa réalisation, ces échanges sont mis en évidence par les ordres (***actions***) et les comptes rendus(***capteurs***), conformément au cahier des charges représentant le bon fonctionnement du système et l'opérateur qui effectue les opérations manuelles.

L'automatisation industrielle et la commande numérique nécessitent l'utilisation des systèmes de contrôle tels que les ordinateurs de commande des machines et les procédés industriels qui réduisent le besoin d'intervention humaine.

En d'autres termes, l'automatisation est l'art d'utiliser les machines afin de réduire la charge de travail des opérateurs humains tout en gardant une productivité et une bonne qualité du produit. Elle fait appel à des systèmes électroniques qui englobent toute la hiérarchie de contrôle-commande depuis les capteurs de mesure en passant par les automates, les bus de communication, la visualisation, jusqu'à la gestion de production et les ressources de l'entreprise. En plus, l'automatisation des applications industrielles joue un rôle de plus en plus important dans l'économie mondiale et dans l'expérience quotidienne.

La compétition technologique et commerciale entre les différents distributeurs mondiaux des composants d'automatisme a accéléré l'évolution des systèmes industriels et des applications automatisées. D'autre part, les tâches exigeant une évaluation et une action subjective, ainsi que les tâches complexes comme la planification stratégique demeurent actuellement en besoin d'une expertise humaine pour qu'elles soient accomplies. Dans de nombreux cas, l'intervention de l'homme s'avère plus judicieuse que les approches mécaniques, même si l'automatisation des tâches industrielles serait possible.

Parmi les principaux processus communément utilisées dans l'industrie on trouve le système de manutention, et plus précisément le système de palettisation d'un produit. Ce dernier fait l'objet de notre étude au sein de l'entreprise IFRI. En effet, afin de suivre et analyser l'architecture du système en terme de fonctions techniques et ses différents synchronisations pour ses divers fonctions, et dans le but d'établir un cahier des charges tenant compte les différentes contraintes de fonctionnement. La nouvelle approche consiste à simuler et visualiser en 3D le processus de fonctionnement à l'aide de l'outil Simulink/Stateflow sous l'environnement MATLAB.

Le present mémoire est scindé en cinq chapitres. Le premier a été consacré pour la présentation de l'outil 3D dans l'environnement MATLAB avec ***Simulink 3D*** (*toolbox*). Le deuxième décrit l'outil de l'automatisation et de supervision ***Stateflow***, L'objectif du troisième chapitre est, dans un premier temps ,de décrire le système réel à automatiser,ensuite une conception en ***3D*** des différents organes constituant le système. Le cinquième chapitre porte sur la programmation et la simulation d'une partie de la chaîne de production en temps réel avec l'outils Simulink/Stateflow. Enfin, dans le dernier chapitre, une implémentation du modèle Simulink/Stateflow à partir de la carte dSPACE et sa validation en temps réel.

Nous conclurons enfin par une synthèse des principaux résultats.

1

Description de l'outil V-Realm Builder©

1.1 Introduction

Si on veut modéliser la dynamique d'un système ou concevoir une machine, il est toujours utile de pouvoir animer la dynamique du modèle, puisque cette animation permet d'observer tous les degrés de libertés d'un système immédiatement plutôt que regardant le 2D. On outre, les animations peuvent aider à comprendre les résultats finaux d'un modèle dynamique sans entrer dans des dérivations mathématiques prolongées. Deux des principaux outils supérieurs dans le développement dynamique de la modélisation et de commande sont MATLAB® et Simulink®. Le Toolbox VRML du MATLAB® est une solution pour agir l'un sur l'autre avec les modèles de réalité-virtuels des systèmes dynamiques. Elle prolonge les possibilités de MATLAB® et de Simulink® au monde des graphiques de réalité- virtuelle.

1.1.1 Présentation de l'environnement MATLAB®

MATLAB® est un logiciel puissant de calcul scientifique : il permet de traiter des problèmes mathématiques complexes, de faire l'analyse et la visualisation des données dans un environnement logiciel ergonomique et intuitif permettant le développement et le déploiement d'algorithmes. Utilisant au maximum les possibilités du calcul matriciel, ce logiciel offre aux non-informaticiens un moyen rapide et efficace de mettre en oeuvre et de tester leurs propres applications techniques.

Depuis le début, MATLAB® a été largement utilisé par les automaticiens. Il possède en effet tous les outils utilisés en automatique, des fonctions de base de l'automatique fréquentielle aux algorithmes plus évolués de l'automatique non linéaire.

MATLAB® est conforté par une multitude de boîtes à outils (*toolboxes*) spécifiques à des domaines variés. En complément de MATLAB®, l'outil additionnel Simulink® a été proposé pour la modélisation et la simulation des systèmes dynamiques en utilisant une représentation de type schémas blocs[4].

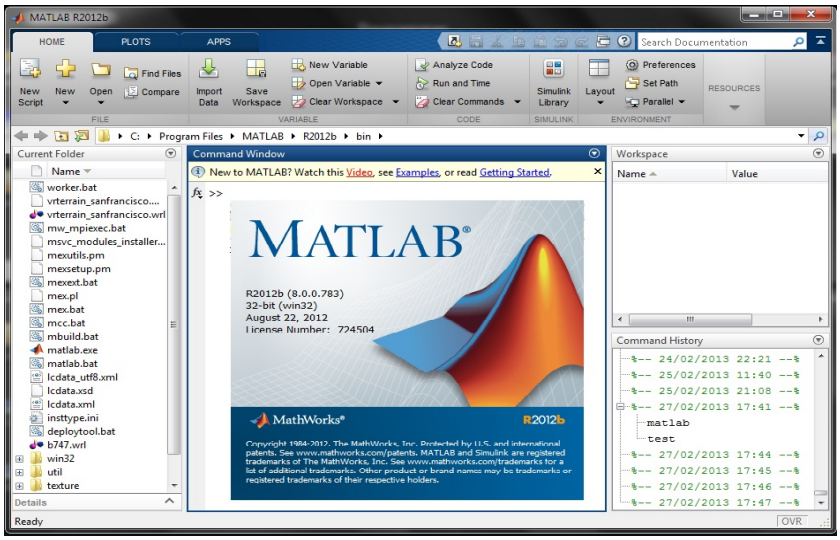


FIGURE 1.1 – *Fenêtre MATLAB (version 8.0)*

1.1.2 Présentation de l'environnement Simulink®

Simulink® est un outil interactif pour la modélisation, la simulation et l'analyse des systèmes multidomaines, continus ou discrets. Son éditeur graphique permet de concevoir rapidement des modèles dynamiques et de simuler leur comportement. Entièrement intégré dans MATLAB®, Simulink® partage avec lui son espace de travail, ce qui permet

d'exploiter avec une grande flexibilité les résultats de simulation [7][8].

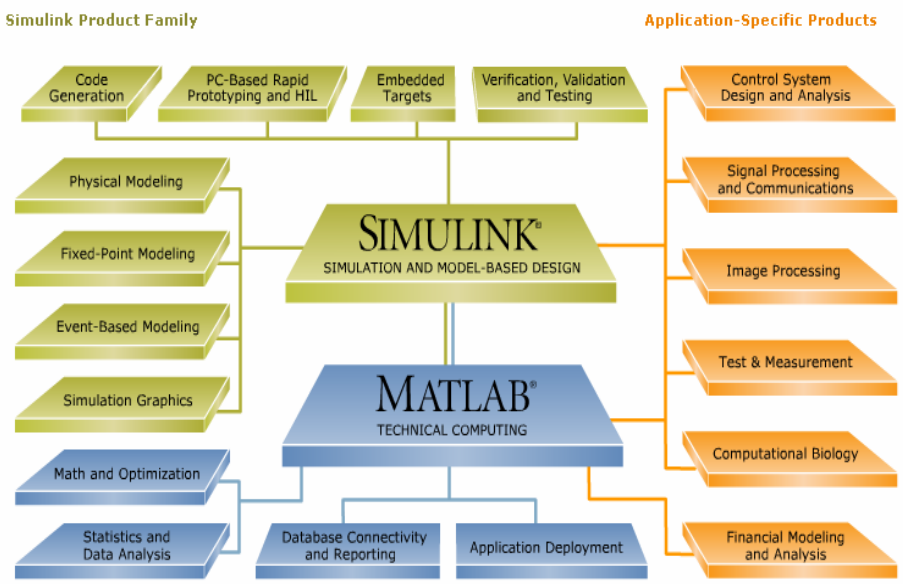


FIGURE 1.2 – *Diagramme illustratif de l'environnement MATLAB/Simulink*

1.1.3 Simulink *3D* animation

L'animation de Simulink *3D* fournit deux rédacteurs pour écrire et importer des mondes de réalité-virtuelle : Constructeur de V-realm builder et l'éditeur du monde *3D*

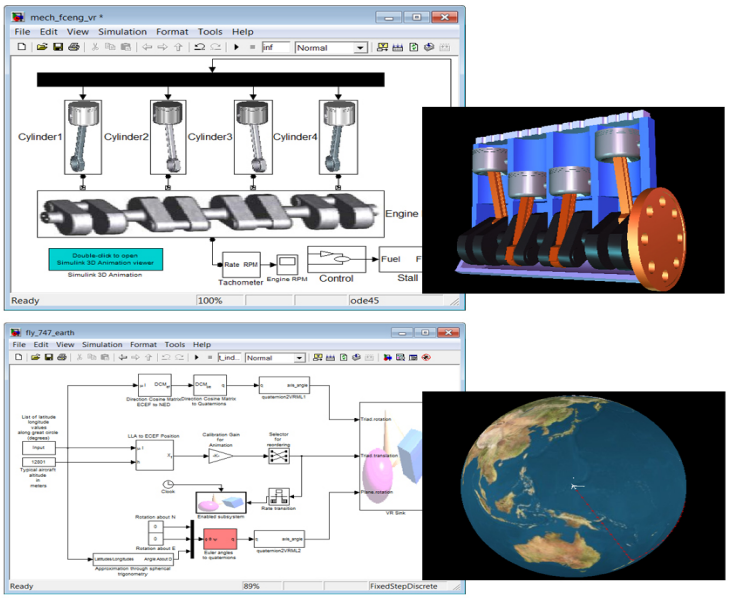


FIGURE 1.3 – *Exemples d'animation en 3D*

1.1.4 L'interface MATLAB via le monde 3D

Avec MATLAB, on peut lire et changer les positions et d'autres propriétés des objets de VRML, lire les signaux des sondes de VRML, créer les rappels de service des outils graphiques, enregistrer les animations. On peut employer MATLAB Compiler pour produire des applications autonomes [5].

1.2 la réalité-virtuelle dans MATLAB et Simulink

1.2.1 Le V-Realm Builder

Le V-Realm Builder parfois désigné sous le nom du VRML est un logiciel qui est employé pour concevoir les mondes virtuels et pour dessiner les objets virtuels en *3D*. Après que l'utilisateur construise son monde virtuel, il peut manoeuvrer les objets et la scène virtuelle par des modèles de commandes de MATLAB et de Simulink pour animer la scène. Les objects de *3D* virtuel ont des propriétés telles que **la translation**, **rotation**, **balance**, et **la couleur** peut être changée par des modèles de commandes de MATLAB et de Simulink afin d'animer la scène[3].

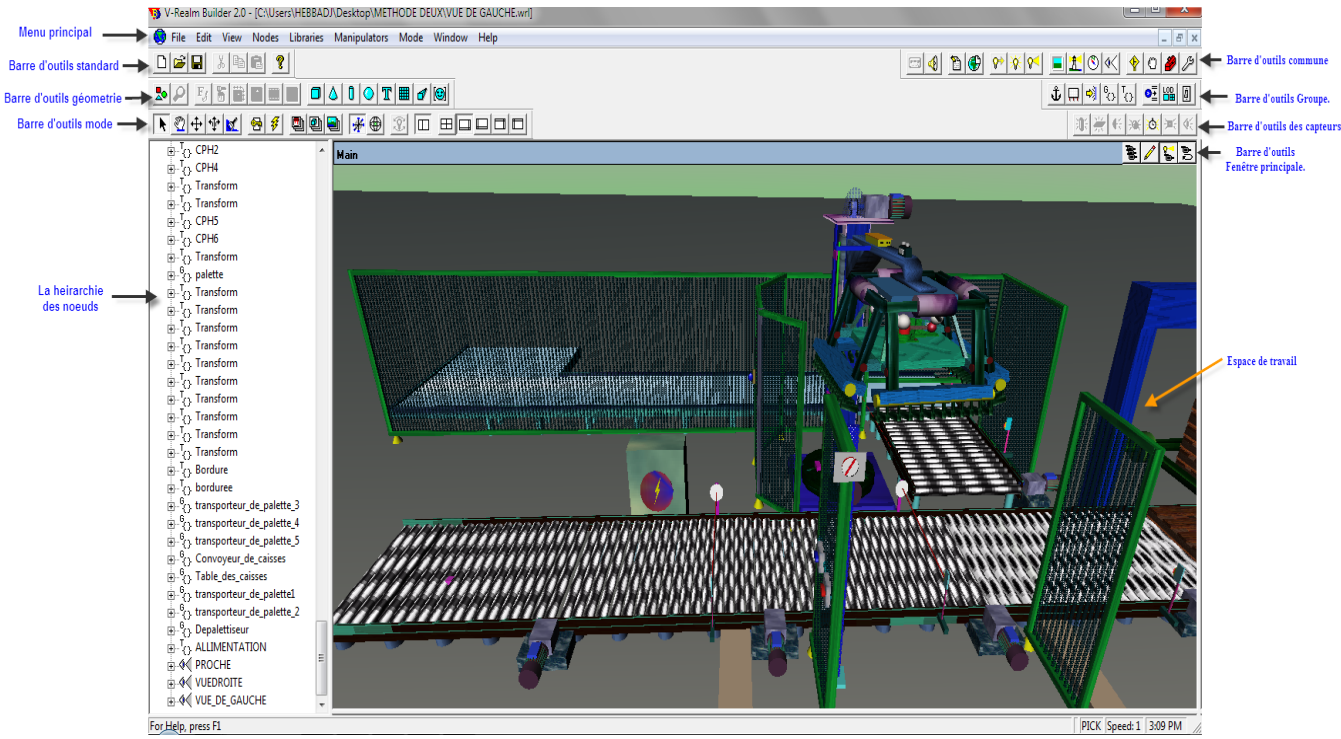


FIGURE 1.4 – *Interface V-Realm Builder*

1.3 Etude plus approfondie de VRML

VRML est un langage basé sur une organisation hiérarchique d'objets (appelés noeuds) ayant un nom et contenant des attributs (appelés champs) dont certains peuvent être modifiés par d'autres objets ou par l'utilisateur. Il existe deux types de noeuds : ceux qui servent à regrouper et opérer la même opération sur tous ses noeuds fils et ceux qui contiennent des données (formes 3D, sons, images, scripts,URL,...).

1.3.1 Les noeuds de regroupement

Les noeuds de regroupement ont un champ pouvant contenir plusieurs autres noeuds. Quel que soit le noeud de regroupement utilisé, ses enfants subissent les mêmes transformations de telle manière que le système de coordonnées des enfants est relatif à celui du noeud de regroupement. On peut alors parler de noeuds parents puisqu'ils transmettent leurs propriétés à leurs enfants. Tous ces noeuds parents peuvent également être inclus dans un autre noeud parent, ce qui a pour effet la création de la hiérarchie sur plusieurs niveaux. Les noeuds parents ainsi que le comportement qu'ils transmettent à leur descendance sont donnés dans le tableau suivant :

Anchor	Si l'utilisateur presse le bouton de la souris alors que le pointeur est sur un objet graphique 3D de la descendance de ce noeud de regroupement alors le navigateur HTML charge l'URL donnée dans le champ url du noeud Anchor.
Billboard	Les axes du système de coordonnées sont modifiés de telle manière que l'axe Z pointe toujours vers le point de vue. Il en résulte quel'utilisateur perçoit le groupe d'objets de la même façon quel que soit son angle de vue.
Collision	il utilise la boite englobante pour les calculs de détection de collision.
Group	Ce noeud sert uniquement à regrouper d'autres noeuds.
LOD	Ce noeud permet de spécifier plusieurs niveaux de détails d'un même groupe d'objets en fonction de la distance à laquelle il se trouve du point de vue.
Switch	il permet d'afficher le groupe d'objets correspondant et pour afficher des objets dont on peut modifier l'état
Transform	Les descendants de ce noeud subissent les transformations spécifiées par les champs rotation, translation, scale et scaleOrientation.

TABLE 1.1 – *Les noeuds de regroupement*

1.3.2 Les noeuds enfants

Les noeuds enfants (que l'on peut ajouter dans la liste des descendants d'un noeud de regroupement) proviennent de 9 familles différentes. Pour chacune d'elles, nous verrons leur utilité, les noeuds la composant et leur description. Le tableau suivant montre les différentes nuances de lumières ainsi leurs descriptions.

1. Les noeuds parents vus précédemment.
2. Les lumières : les trois noeuds servant à définir les trois types de lumières utilisables dans une scène VRML. Chacune d'elles comporte les champs intensity (donnant l'intensité, l'éclat), color (donnant la couleur de la lumière) et ambientIntensity (donnant l'intensité de la lumière ambiante diffusée par cette lumière). Elles diffèrent l'une de l'autre par le type d'éclairage :

DirectionnalLight	Illumine tous les descendants du noeud qui la contient.
PointLight	Illumine tous les objets de la scène qui se trouvent dans sa zone d'influence sphérique.
SpotLight	Illumine tous les objets de la scène qui se trouvent dans sa zone d'influence conique.

TABLE 1.2 – *Les noeuds de lumières*

Les détecteurs : les 7 noeuds suivants servent à déterminer le type d'événements qui doit être récupéré. A ces sept noeuds peut être rajouté le noeud Anchor puisqu'il définit également un événement. Mais du fait qu'il englobe d'autres noeuds, nous préférons le classer parmi les noeuds de regroupement.

CylinderSensor	Evénement activé lorsque le pointeur de la souris passe sur un descendant du noeud parent contenant le sondeur et que les coordonnées 3D de la souris, transformées en coordonnées cylindriques, pointent dans une zone cylindrique définie.
PlaneSensor	Evénement activé lorsque le pointeur de la souris passe sur un descendant du noeud parent contenant le sondeur et que les coordonnées de la souris, projetées sur un plan, pointent dans une zone rectangulaire définie.
ProximitySensor	Evénement activé lorsque le point de vue entre, sort ou est déplacé dans la zone définie par un pavé.
SphereSensor	Evénement activé lorsque le pointeur de la souris passe sur un descendant du noeud parent contenant le sondeur et que les coordonnées du pointeur, en coordonnées sphériques, sont dans une zone conique définie.
TimeSensor	Evénement activé soit à un moment précis .
TouchSensor	Evénement activé lorsque le pointeur de la souris passe sur un descendant du noeud parent contenant le sondeur.
VisibilitySensor	Evénement activé lorsqú'une zone définie par un pavé change d'état (devient visible ou invisible) en fonction des mouvements du point de vue.

TABLE 1.3 – *Les noeuds de capteurs*

Les noeuds liés : on peut dénombrer quatres noeuds qui sont gérés dans une pile afin qu'uniquement un de chaque sorte puisse être actif à un moment donné mais que l'on puisse les changer.

Background	Définit le type de fond : couleur, texture.
Fog	Utilisé pour simuler des effets atmosphériques.
NavigationInfo	Contient des informations sur la navigation comme le type (marcher, voler), la vitesse, la limite de visibilité.
Viewpoint	Définit différents points de vue sur la scène.

TABLE 1.4 – *Les différents types de noeuds liés*

Les objets graphiques et sonores : ces deux objets sont les seuls permettant d'ajouter dans la scène des objets perceptibles par l'utilisateur.

Shape	Permet d'ajouter un objet graphique 3D. Ce noeud est composé de deux champs attendant d'autres noeuds (dépendants). Le champ geometry doit référencer un noeud de géométrie et le champ appearance doit toujours référencer un noeud de type Appearance.
Sound	Permet d'ajouter un son à la scène. Son champ source référence soit un noeud AudioClip soit un noeud MovieTexture.

TABLE 1.5 – *Les objets graphiques et sonores*

1.3.3 Les noeuds dépendants

Le troisième type de noeuds, sont les noeuds qui ne peuvent pas être ni des noeuds parents ni contenus dans un noeud parent. Cependant, ces noeuds peuvent, en effet, être utilisés qu'en attribut d'un autre noeud. Ce sont des noeuds que nous appellerons dépendants, donnant les formes géométriques, les propriétés des formes géométriques, les apparences et les propriétés des apparences. Les noeuds dépendants (noeud + champ entre parenthèses) ainsi leurs caractéristiques(utilité) sont regroupés dans le tableau suivant.

Appearance	Shape (appearance)	Regroupe les options d'apparence de la forme 3D définie dans le champ geometry du noeud Shape.
AudioClip	Sound (source)	Donne l'URL du son à jouer par le noeud Sound.
Box	Shape (geometry)	Noeud créant une boite.
Cone	Shape (geometry)	Noeud créant un cône
Cylinder	Shape (geometry)	Noeud créant un cylindre.
Extrusion	Shape (geometry)	Noeud créant une forme 3D par extrusion.
ImageTexture	Appearance (texture)	Permet de spécifier une texture fixe pour l'objet
Material	Appearance (material)	il donne les propriétés du matériel d'un objet.
Sphere	Shape (geometry)	Noeud créant une sphère.
Text	Shape (geometry)	Noeud créant un texte.

TABLE 1.6 – *Les noeuds dépendants*

1.3.4 Le format de fichier VRML

Les fichiers VRML portent l'extension .wrl (abréviation de world).

1.3.5 Le graphe de scène

Le graphe de scène présentant la hiérarchie des noeuds constitue la troisième partie d'un fichier VRML. Le schéma suivant correspond au graphe de scène

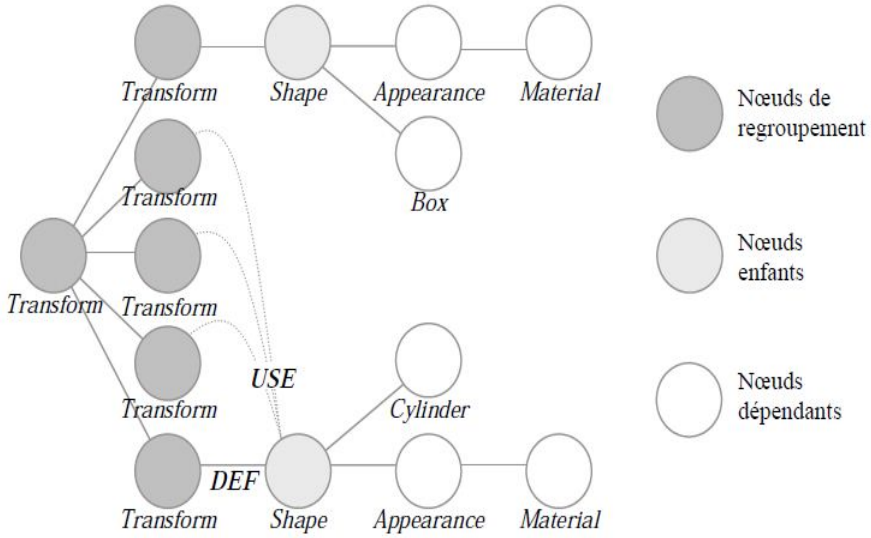


FIGURE 1.5 – *Exemple de graphe de scène VRML.*

1.3.5.1 Propriétés principales du MATLAB VRML Toolbox

On distingue quatre propriétés principales :

- Il permet de créer les mondes virtuels ou les scènes tridimensionnelles en utilisant la réalité-virtuelle standard modelant la technologie de la langue (VRML) ;
- création et définition des systèmes dynamiques avec MATLAB et Simulink ;
- visualiser les scénarios tridimensionnels en mouvement généré par des signaux de l'environment de Simulink ;
- changer les positions et les propriétés des Objets dans un monde Virtuel au cours d'une simulation.

Le VRML Toolbox fonctionne avec MATLAB(script) et Simulink. Cependant, l'interface Simulink est la manière préférée de travailler avec le toolbox.

1.3.5.2 Association d'un monde virtuel à Simulink

Avec le VRML MATLAB Toolbox nous pouvons connecter un schéma fonctionnel de Simulink avec un monde virtuel[1]. La simulation d'un modèle de Simulink produit des données de signal pour un système dynamique. En reliant le modèle de Simulink à un monde virtuel, nous pouvons employer ces données pour commander et animer le monde virtuel.

1.4 conclusion

Dans ce chapitre, nous avons essayé de présenter le logiciel, par la définition de différentes caractéristiques, et les principales fonctions du logiciel V-Realm Builder et la norme du langage VRML.

Ce travail nous a permis de bien comprendre notre outil de conception, et cela pour mieux développer l'environnement virtuel du dépaléttiseur afin de le valider, au moins du point de vue du comportement dynamique de notre système à simuler.

2

Description de l'outil Stateflow®

Stateflow est un outil interactif de conception de **systèmes événementiels**. Basé sur la théorie des machines à états finis, il permet de concevoir graphiquement des systèmes de logique de supervision ou de contrôle. Entièrement intégré à Simulink, il complète efficacement cet environnement de simulation. C'est un *Toolbox* récent de Simulink apparu avec la 5^{eme} version de MATLAB (Simulink 4). Il est considéré comme outil de description qu'on peut comparer au ***grafcet***, lequel permet de décrire, sous une forme normalisée la logique du comportement des **systèmes à événements discrets.**[9]

Un système à événement discret est un système dynamique dans lequel l'espace des états est discret. Ses trajectoires d'états sont constantes par morceaux. Un tel système évolue conformément à l'occurrence des événement physiques à des intervalles de temps généralement irréguliers ou inconnus.

Stateflow fonctionne en symbiose avec Simulink sous MATLAB. Il est donc possible de modéliser un processus industriel et de simuler l'enchaînement des tâches en même temps quelle s'execute. Le comportement d'un processus physique automatisé est décrit dans un document qui définit les **étapes** du fonctionnement et leur enchaînement. Il précise les **conditions** de passage d'une étape à l'autre et les **actions** quelle doivent exécuter. Ce

descriptif en forme de cahier des charges, se présente en principe sous une forme normalisée . Stateflow représente lui aussi la partie commande d'un ensemble automatisé par un diagramme état-transition. Il permet de simuler le fonctionnement du diagramme. Il est donc possible en utilisant Simulink pour la partie ***opérative*** et Stateflow pour la partie ***commande*** de simuler complètement une machine automatisée. Stateflow pourra même assurer la génération et l'implantation du code exécutable dans un organe de traitement (*calculateur,automate programmable ...*)[6].

2.1 Machine à états finis

Une machine à états finis est un modèle décrivant le comportement d'un nombre fini d'états, des transitions entre ces états et d'actions.

Une machine à états finis peut être représentée par un diagramme d'état ou une table de transition d'état[4].

Lorsqu'on clique sur le bloc Chart dans la fenêtre Simulink, on aboutit à une fenêtre Stateflow dans laquelle nous trouvons une palette d'outils nécessaires pour construire une machine d'état fini.

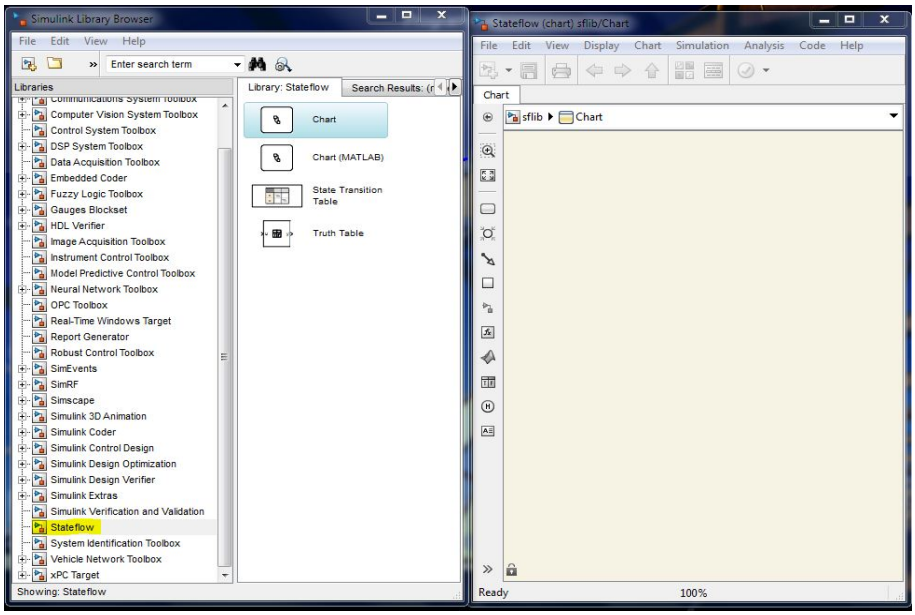


FIGURE 2.1 – *Fenêtre Stateflow.*

Ces outils sont représentés dans le tableau suivant :

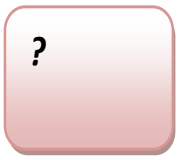
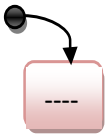
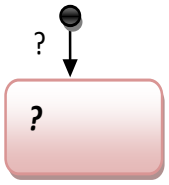
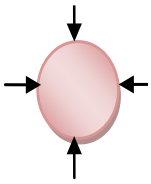
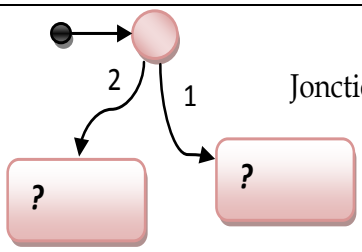
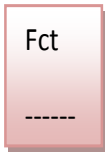
	 Rectangle qui définit l'état à qui l'on donne un nom à la position du signe <?>
	History junction ou jonction de l'historique ou est Enregistrée l'historique d'activé d'un état.
	 Transition d'un état à un autre sous certaine Condition que l'on écrit à la position du <?>
	 Jonction ou arrivent et partent plusieurs Transitions. Les transitions sont exécutées, par défauts selon l'ordre de leur création .
	 Etat spéciale appelé <Table de vérité>
	Fonctions graphiques
	Box
	 Fonction MATLAB et Simulink (Embedded function)

FIGURE 2.2 – *Outils de Stateflow*

2.2 Objets de Stateflow

Un graphe Stateflow est composé des différents objets suivants :

- États,
- Événements,
- Transition,
- Les données
- Les conditions,
- Jonction,
- Jonction de l'historique,
- etc.

2.2.1 Les états

Dans les états on décrit les actions à réaliser dans le cadre du graphe qui décrit la machine à états finis. Un état peut être actif ou inactif selon des événements qui peuvent intervenir ou des conditions de validation des transitions.

Chaque état ayant des sous états peut être décomposé soit en parallèle (***les sous états s'exécutent simultanément***) ou exclusif (***les sous états s'exécutent exclusivement***).

Chaque état possède un nom (label) en haut à gauche du rectangle délimitant cet état.

nom/

entry : actions à réaliser à l'entrée de cet état

during : actions à réaliser Durant l'activité de cet état

exit : actions à réaliser dès qu'on sort de cet état

on <nom-evt> : actions à réaliser à l'occurrence de l'événement nom-evt

bind : <variable> <actions>

2.2.2 Les transitions

Une transition est un objet graphique qui lie un état à un autre. Un label décrit les circonstances ou les conditions du passage entre ces états.

2.2.3 Les transitions par défaut

La transition par défaut (sans aucune condition, ni label) détermine l'état qui doit être actif lorsqu'il y a ambiguïté entre plusieurs états en décomposition "OU" et qui ont le même niveau hiérarchique.

Chaque état qui contient des sous-états doit avoir sur l'un d'eux une transition par défaut qui détermine cet état actif dès l'entrée de ce super état (ou état parent).

2.2.4 Les événements

Les événements, objets non graphiques, agissent sur l'exécution du graphe Stateflow. Tous les événements doivent être définis par le menu Add/Event dans Model Explorer. L'occurrence d'un événement peut servir à valider une transition de passage d'un état à un autre ou une action à exécuter.

2.2.5 Les objets Data

Les objets Data (données) peuvent être des variables locales, 'entrées de' ou 'sorties vers' Simulink, une constante, etc.

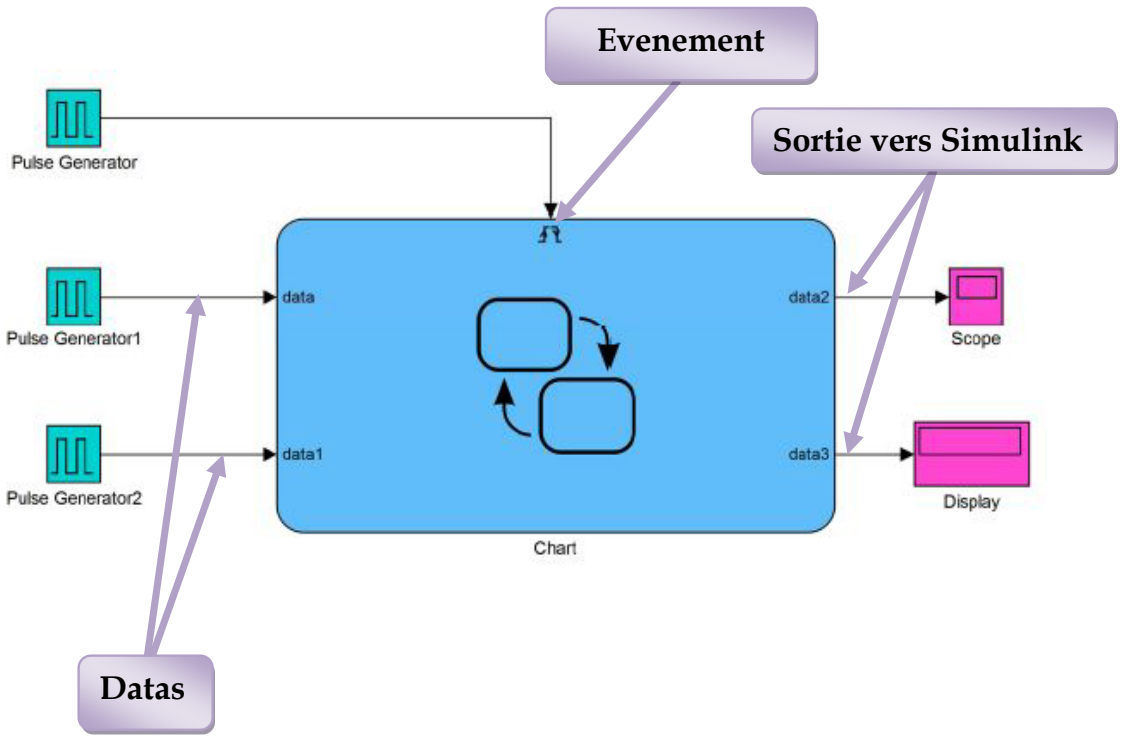


FIGURE 2.3 – *Exemple d'un schema hybride Simulink/Stateflow*

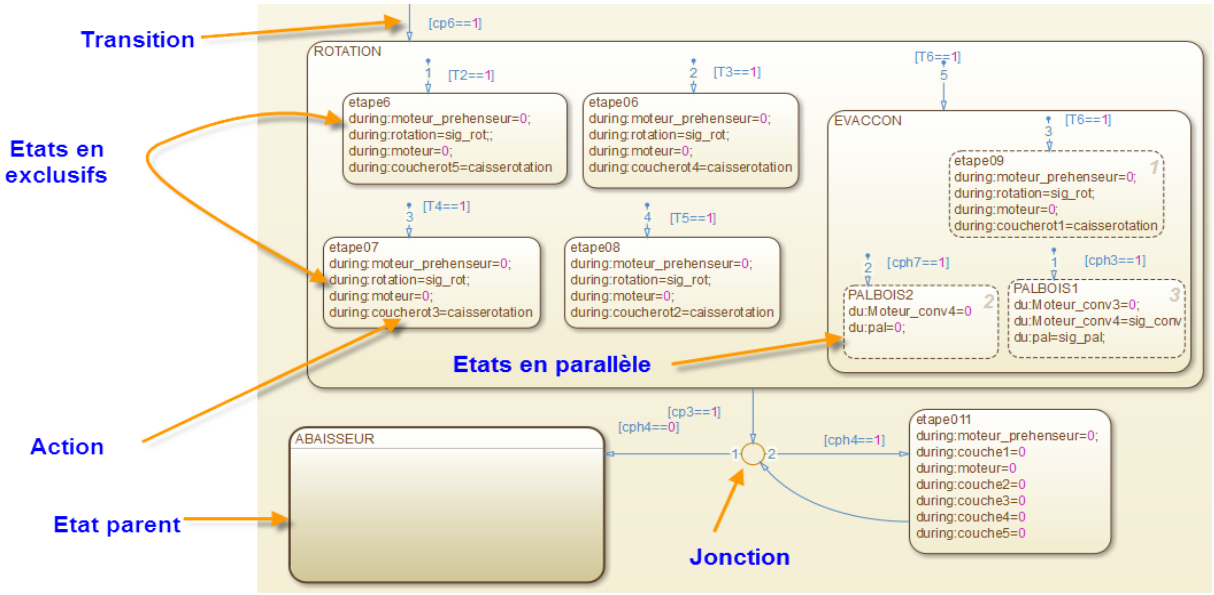


FIGURE 2.4 – *Exemple d'un diagramme Stateflow*

2.3 Procédure pour créer un Stateflow

Pour créer un diagramme de Stateflow, on présente le schéma fonctionnel qui est sur le schéma ci-dessous [2] :

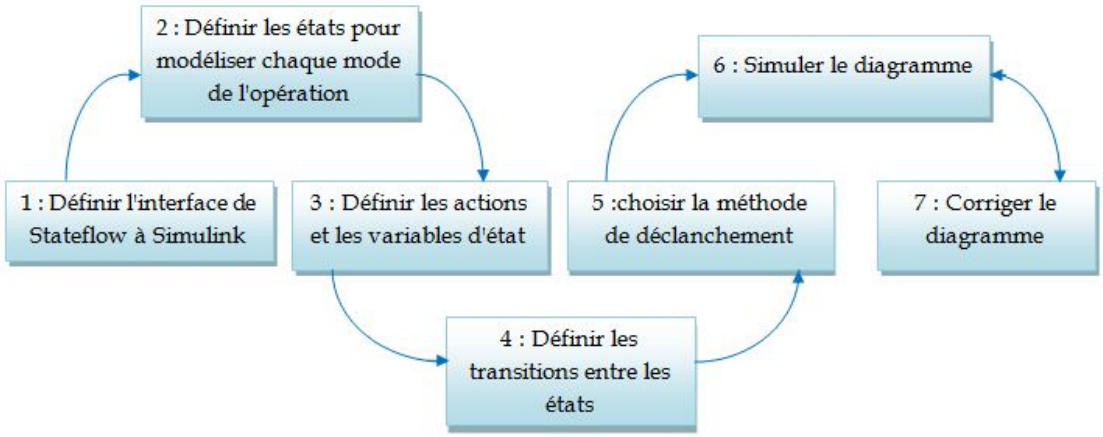


FIGURE 2.5 – *Procédure recommandée pour créer un modèle de Simulink avec un diagramme de Stateflow*

2.4 La terminologie

Un schéma stateflow doit être conçu de manière hiérarchisée [6]. Les objets graphiques et non graphiques vont s'emboîter selon la hiérarchie parent-enfants.

Le plus haut niveau de la hiérarchie stateflow est appelé (***machine***) ; il correspond au niveau(***modèle***) dans la hiérarchie.

Le diagramme(*chart*) : Ce deuxième niveau, correspond au sous système Simulink, décrit une partie de la machine. Il rassemble des ***états*** reliés par des ***transitions*** dont il est le parent.

L'état(*State*) : Ce niveau correspond à l'état qu'on décrit par un vecteur d'état dans simulink. Il représente l'***état*** dans lequel se trouve la machine, en quelque sorte l'état actuel de son fonctionnement.

Un ***état*** peut contenir des ***sous-état*** qui sont ses enfants, il devient alors un ***super-état***.

2.4.1 Présentation d'un exemple illustratif

Afin de permettre la présentation des principes et des règles nécessaires pour la mise au point d'un diagramme sous stateflow, nous présentons l'exemple de mode de marche/arrêt d'un chauffe-eau ; le système bascule entre deux états. Il est en état de marche noté 'M' lorsque l'interrupteur est en position de marche et que la température de l'eau est inférieure à 30 degrés Celsius. Sinon il reste dans l'état arrêt noté 'A' tant que la température est supérieure à 30 degrés Celsius. A partir de l'état de marche, le système peut revenir à l'arret lorsque l'interrupteur en position arret. Ceci est illustré dans la figure suivante :

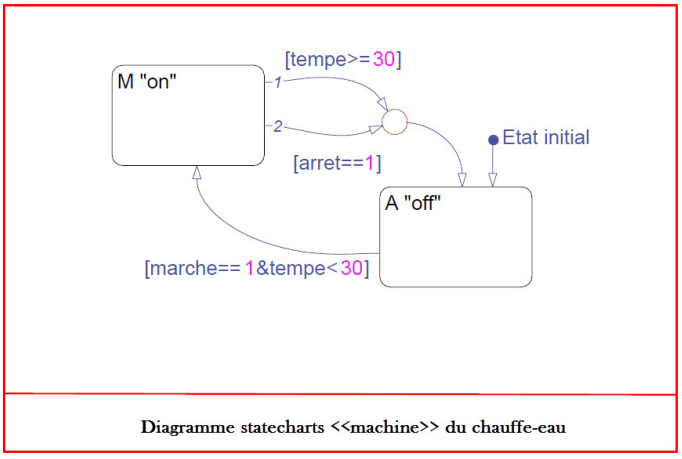


FIGURE 2.6 – *Modèle de fonctionnement du chauffe-eau*

2.5 Conclusion

Les avantages de cet outil résident dans le fait qu'il permet de prendre en compte les aspects de parallélisme et d'hiérarchie de façon naturelle, il permet aussi de réduire le nombre d'arcs entre les différents états du système, lorsqu'il s'agit d'une entrée vers un état composé (état parent contenant des sous-états) et aussi lors d'une transition d'un état composé vers un autre. Nous verrons ceci en détails dans l'exemple d'application au cours du fonctionnement en temps réel.

3

Conception en 3D de la chaîne à automatiser

3.1 Introduction

La simulation d'un modèle de simulink produit des données de signal pour un système dynamique. En reliant le modèle de Simulink à un monde virtuel, nous pouvons employer ces données pour commander et animer le monde virtuel.

Créer un monde virtuel revient à trouver ou établir un objet de trois dimensions. Lier ce monde virtuel à MATLAB ou Simulink pour l'analyse de chaque étape qui porte ses propres défis.

3.2 Description détaillé du système

Avant toute démarche de la simulation, il importe d'étudier le système aussi bien sur l'aspect structurel que sur le l'aspect fonctionnel.

3.2.1 Description de Transport de palettes

Le système de transport de palettes comprend différentes unités de transport qui peuvent être reliées entre eux selon le principe de construction modulaire. Cela permet, selon le but d'utilisation et la mise en place de l'installation, le transport de différents types de palettes (par ex. palettes pleines ou vides).

Le transfert et la déviation des palettes, d'une unité de transport à l'autre, se passe de manière entièrement automatique par l'intermédiaire d'une commande séquentielle.

3.2.1.1 Transporteur à rouleaux

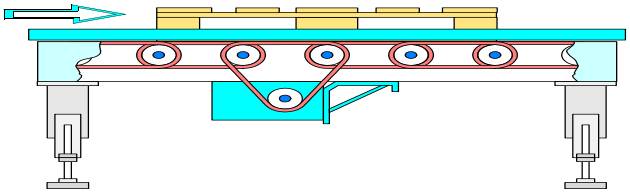


FIGURE 3.1 – *Convoyeur de palettes*

Les rouleaux sont entraînés individuellement par l'intermédiaire de chaînes. Les unités d'entraînement sont protégés par des tôles de protection.

3.2.2 Description du magasin de stockage de palettes

Le magasin de stockage de palettes est un magasin intégré dans le transporteur de palettes vides. Suivant l'exécution, le magasin de stockage de palettes peut être utilisé de façon individuelle :

- comme tampon pour l'empilage et le désempilage combinés
- comme magasin d'empilage (par ex. pour palettes défectueuses)

Le magasin de stockage de palettes est entièrement automatique. Il est, en outre, équipé d'une commande manuelle à partir de laquelle toutes les exécutions des mouvements peuvent se faire manuellement. le tableau suivant rassemble les différentes zones et éléments de construction de la machine.

Zone de la machine	**Élément de construction**
Mécanisme de déplacement vertical (1)	(a) Entraînement avec chaînes de levage (b) Cadre de levage
Mécanisme de déplacement transversal (2)	(a) Entraînement avec chaînes d'entraînement (b) Barre de réception de palettes
Transporteur de palettes (3)	Transporteur de palettes
Dispositif de sécurité (4)	(a) Colonne pour cellule photoélectrique (b) Cellule photoélectrique

TABLE 3.1 – Les zone et les elements de construction de la machine

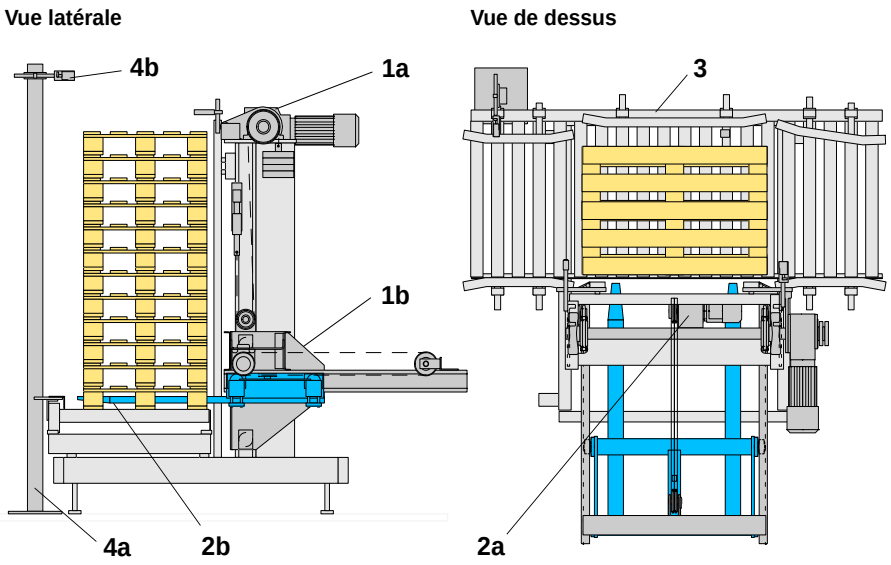


FIGURE 3.2 – *Schéma conceptuel du magasin de stockage de palettes*

3.2.3 Mode de fonctionnement de transport de palettes

Chaque unité de transport est équipée de dispositifs de commande.

Le transport de palettes est commandé par une commande séquentielle, c-à-d une fois chargée, la palette est acheminée automatiquement d'une unité à l'autre.

Le transfert d'une palette d'une unité à l'autre (commande de l'entraînement des unités de transport) est effectué par l'intermédiaire des dispositifs de commande.

3.2.4 Les dispositifs de commande

Le capteur est le composant qui permet d'informer la partie commande d'un état de la partie opérative ou du milieu extérieur. Il prélève une information sur la grandeur physique de la partie opérative puis la convertir en une information exploitable par la partie commande.

Dans la structure d'un système automatisé, le capteur fait partie de la chaîne d'acquisition. Les capteurs qui délivrent un signal binaire(0 ou 1) sont appelés des détecteurs,ou capteur TOR (Tout Ou Rien).

parmi les capteurs existants on trouve :

ı) **:Cellules photo-électriques :** Les cellules photoélectriques sont activées par le profil latéral de la palette.

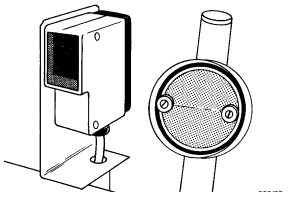


FIGURE 3.3 – *Cellule photo-électriques*

Principe :

Les détecteurs photo-électriques se composent essentiellement d'un émetteur de lumière associé à un récepteur photo-sensible. L'émetteur et le récepteur sont dans le même boîtier.

Le faisceau lumineux émis et renvoyé vers le récepteur par **un réflecteur**. La détection se fait par coupure de faisceau.

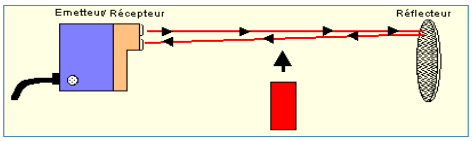


FIGURE 3.4 – *Cellule photo-électriques Reflex*

Utilisation : Détection de tout objet opaque.

Avantages :

– Pas de contact physique avec l'objet détecté ;

– Pas d'usure ; possibilité de détecter des objets fragiles ;
– Détection sur de grande distance ;
– généralement en lumière infrarouge invisible, indépendante des conditions d'environnement.

ıı) **:Interrupteur à galets :** Les interrupteurs à galets sont activés par la planche inférieure de la palette.

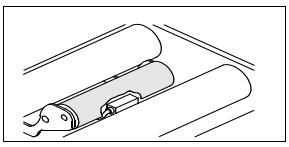


FIGURE 3.5 – *Interrupteur à galet*

Il peuvent être appelés :
– Détecteur de position ;
– Interrupteur de position mécanique ou fin de course.

Les interrupteurs de position mécanique coupent ou établissent un circuit lorsqu'ils sont actionnés par un mobile.

La tête de commande provoque le basculement d'un contact électrique qui délivre le signal de sortie. Ces capteurs sont dits à contact direct.

Avantages :
– fiabilité de contact ;
– bonne fidélité sur les points d'enclenchement (jusqu'à 0,01 mm) ;
– mise en oeuvre simple et fonctionnement visualisé ;
– grande résistance aux ambiances industrielles.

ıı) **:Detecteur de proximité inductif :** Les détecteurs inductifs détectent exclusivement les objets métalliques. Ce sont des capteurs électroniques statiques détectant sans contact physique la présence d'un élément. Ils permettent de détecter des objets métalliques à une distance variable de 0 à 60mm

Fonctionnement : Le détecteur émet un champ électromagnétique, une pièce métallique située dans ce champ en modifie les caractéristiques, le détecteur transforme cette information en signal électrique de sortie.

Caractéristiques :
– Durée de vie importante ;

– supporte bien les ambiance humides et poussiéreuse.

3.2.5 Description de Dépalettiseur "Universel Pressant 2500 1N"

Ce dépalettiseur permet de charger ou décharger des caisses sur la palette avec une grande précision. Cette machine a une colonne de levage tournable et elle peut être équipée de différents outils. Le Pressant 2500 1N peut être employé comme ***palettiseur*** pour les caisses. Pour ce faire, les caisses soient d'abord groupées en couches à une station de groupement. Après, la couche complète sera soulevée et retournée au-dessus de la palette par l'intermédiaire d'un moteur qui fait le tour de 90° et placée sur la palette. Utilisé comme ***dépalettiseur***, le Pressant 2500 1N traite exclusivement les paquets restituables. Dans ce cas, la tête de la pince soulève la couche des caisses de la palette et l'emporte à la table de décharge avec un tour de 90°.

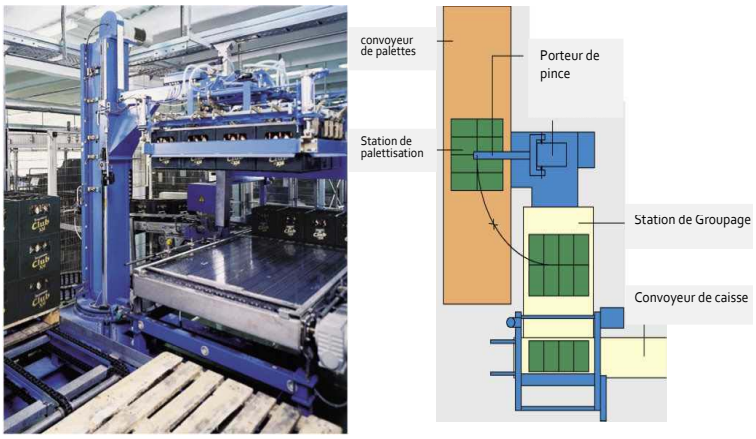


FIGURE 3.6 – *Dépalettiseur Pressant 2500 1N*

3.2.5.1 Données générales

Branchements principaux :

*a)***Électrique :**

Voltage	220/380 V 5
Fréquence	50 Hz 2
Puissance apparente	27 KVA
Puissance active	26,5 KW

*b)***Pneumatique :**

Pression 5 bar

Point de condensation -5◦ C

Filtre 40 μm

Poids 5000 kg

Charge au sol max. 8 x 8000 N

Niveau sonore $\leq$85 dB

3.2.6 Spécification des différents sous-ensemble de la machine

Cadres inférieur : Construction robuste de support en I, toutes les parties sont peintes avec une peinture à 2 composants de grande qualité, tous les conducteurs pneumatiques et électriques sont installés dans des conduites avec couvercles faciles à ouvrir et fermer.

Colonnes : La construction en tubes de grandes dimensions garantit un fonctionnement de la machine avec une sécurité de service élevée. Les glissières sur colonnes vissées et durcies sont facilement remplaçables et offrent une résistance à l'usure élevée.

Élévateur : Composé d'une unité d'entraînement avec courroies dentées et contrepoids. Équipé d'un motoréducteur de frein à engrenages cylindriques avec régulation de fréquence qui est relié, par un arbre articulé, à un réducteur individuel à engrenages cylindriques.

L'unité d'entraînement permet, par l'utilisation d'un convertisseur de fréquence, une précision de positionnement élevée et une dynamique de déplacement optimale du mécanisme de déplacement vertical.

convoyeur de caisses : Table d'évacuation en matériau résistant à la corrosion, équipée de chaînes à charnière. Une cellule photo-électrique permet de contrôler l'espace libre. L'entraînement s'effectue au moyen d'un moteur à courant triphasé.

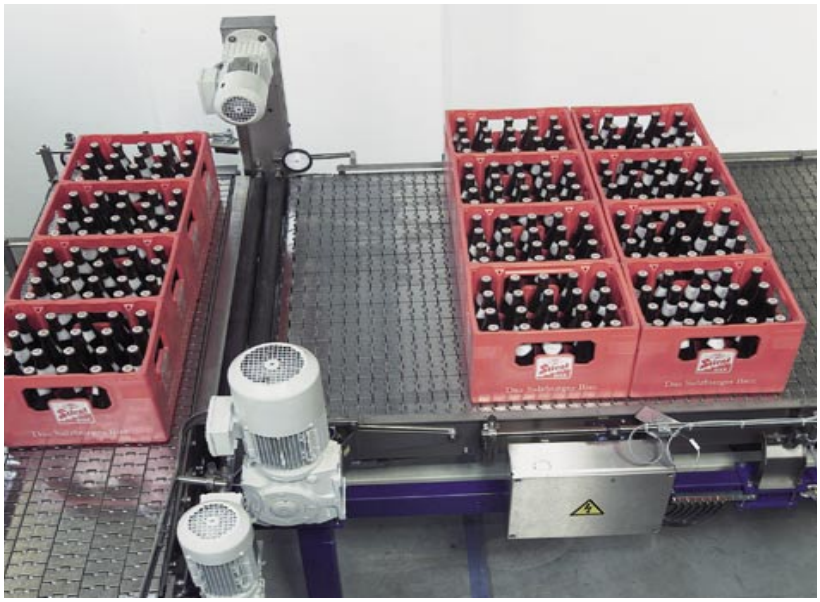


FIGURE 3.7 – *Table des caisses*

Dispositif de sécurité : les grillages de sécurité avec cellules photo-électrique de sécurité.

FIGURE 3.8 – *Grillage de sécurité*

3.3 Modèle virtuel d'un Dépaléttiseur

Le simulateur virtuel que nous avons développé était fondamentalement l'integration entre le logiciel V-realm Builder et le toolbox de VRML,nous avons crée le monde virtuel entier de Dépaléttiseur avec le transporteur de palettes virtuels en utilisant le rédacteur à

trois dimensions de V-realm Builder et alors, nous l'avons exporté dans le VRML Toolbox de MATLAB.

Les étapes que nous avons suivies pour le développement du simulateur virtuel sont enchaînées comme suit :

3.3.1 Conception virtuelle du Dépaléttiseur

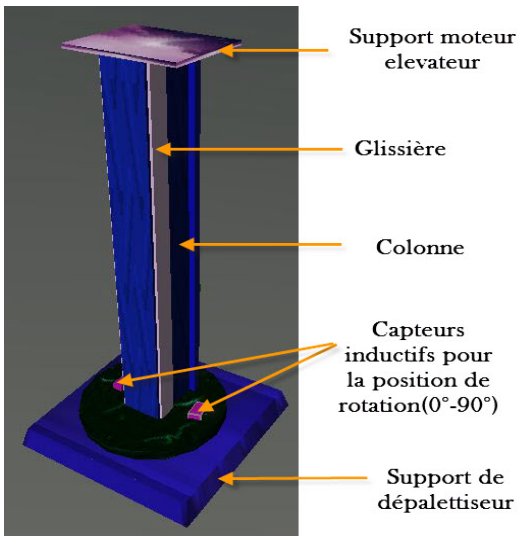


FIGURE 3.9 – *Colonnes et glissières*

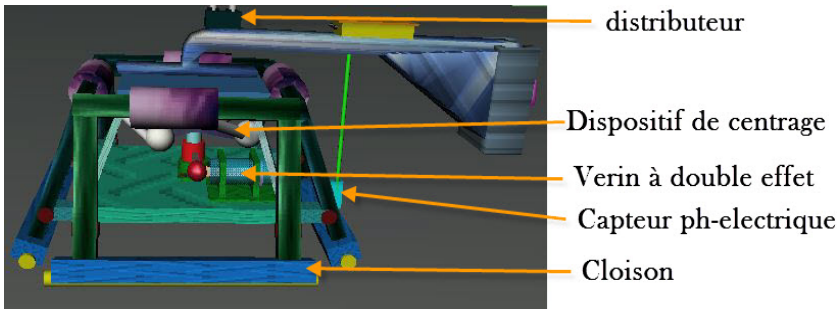


FIGURE 3.10 – *La Pince*

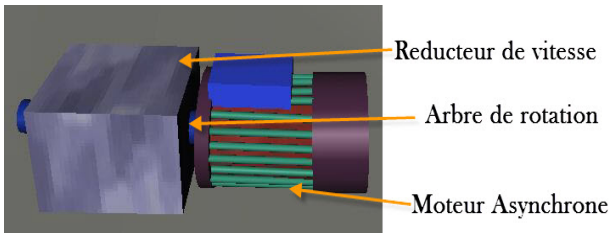


FIGURE 3.11 – *Moteur 3ϕ avec réducteur*

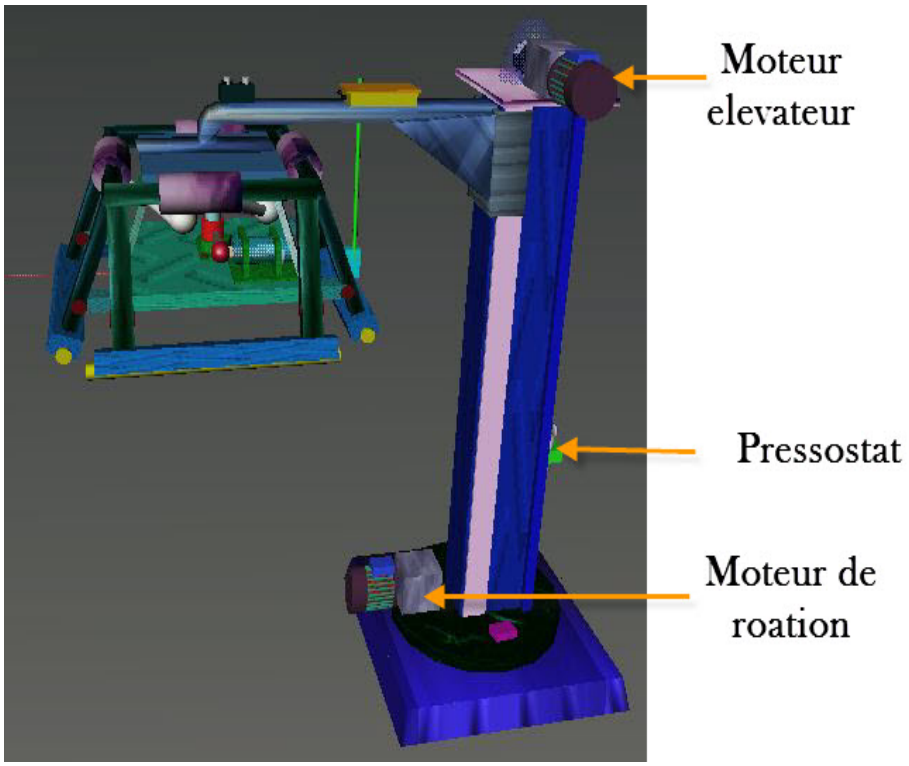


FIGURE 3.12 – *Le dépalettiseur 1N*

3.3.2 Conception virtuelle du Transport de palettes

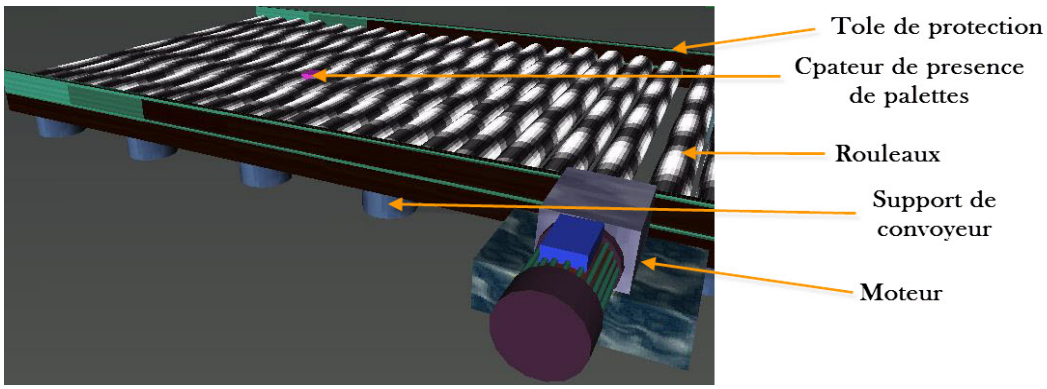


FIGURE 3.13 – *Convoyeur de palettes*

3.3.3 Conception virtuelle d'une palettes pleine

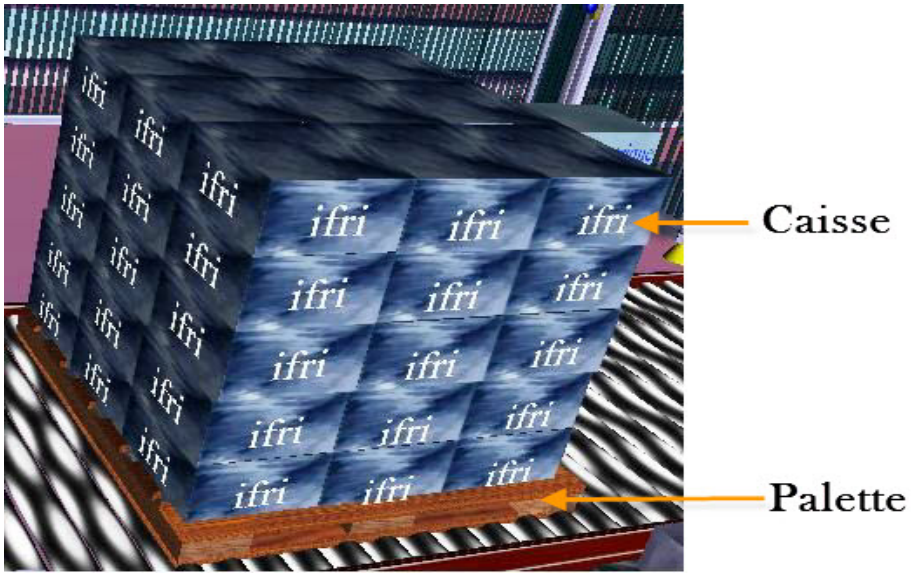


FIGURE 3.14 – *Palette à cinq couches*

3.3.4 Integration entre le le Dépaléttiseur et le transporteur de palette

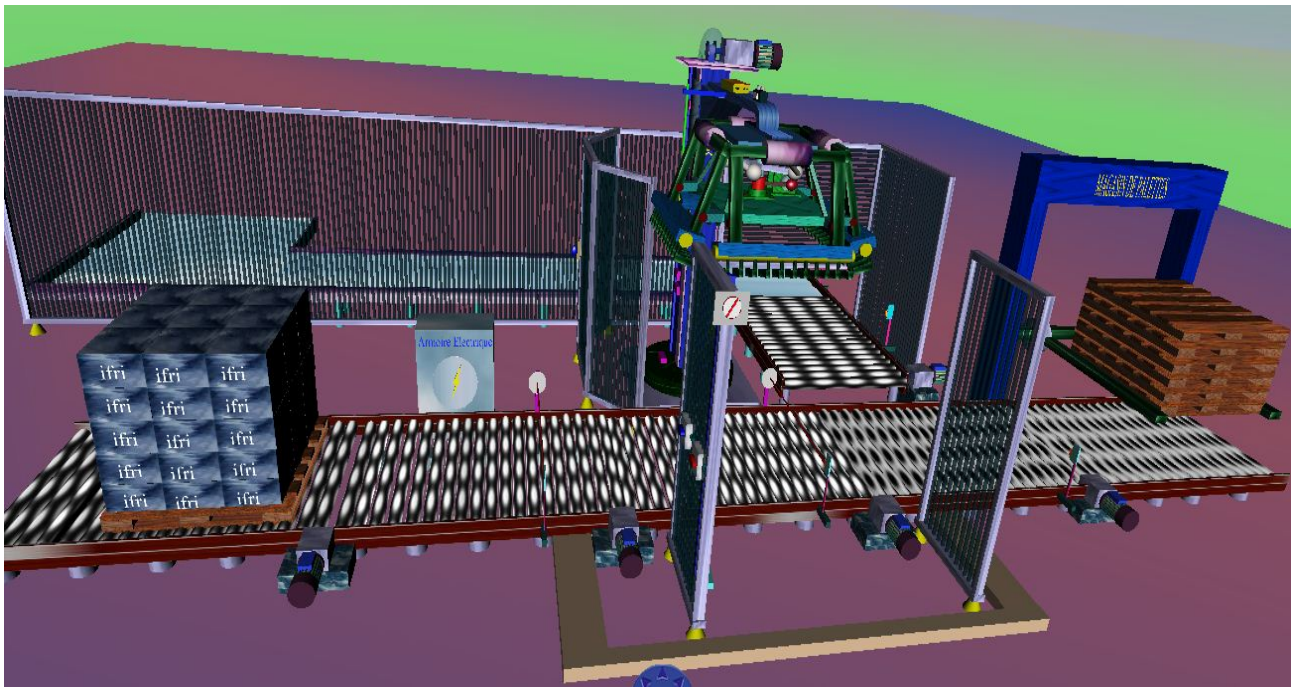


FIGURE 3.15 – *Dépalettiseur avec Transport de palettes*

3.3.5 Définition des paramètres dans le monde virtuel

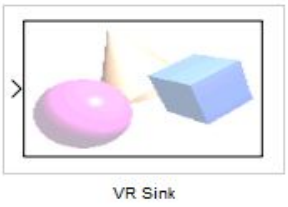


FIGURE 3.16 – *VR Sink*

VR Sink : Écrit les données de modèle de Simulink au monde virtuel, des valeurs de ses ports au gisements virtuels du monde indiqués dans la zone de dialogue de paramètres de bloc.

3.3.5.1 Création d'un modèle Simulink pour animer une scène virtuelle

Les étapes à suivre afin de créer une scène virtuelle sous Simulink sont les suivantes :

1. Créer un nouveau modèle Simulink (*Fig.3.17*) ;
2. Ajouter le bloc **VR.Sink** au nouveau modèle à partir de la bibliothèque de Simulink 3D ;
3. Charger le fichier '*wrl*' dans le bloc **VR.Sink** qui contiendra la scène virtuelle afin de manoeuvrer la scène par les entrées de Simulink (*Fig.3.18*) ;

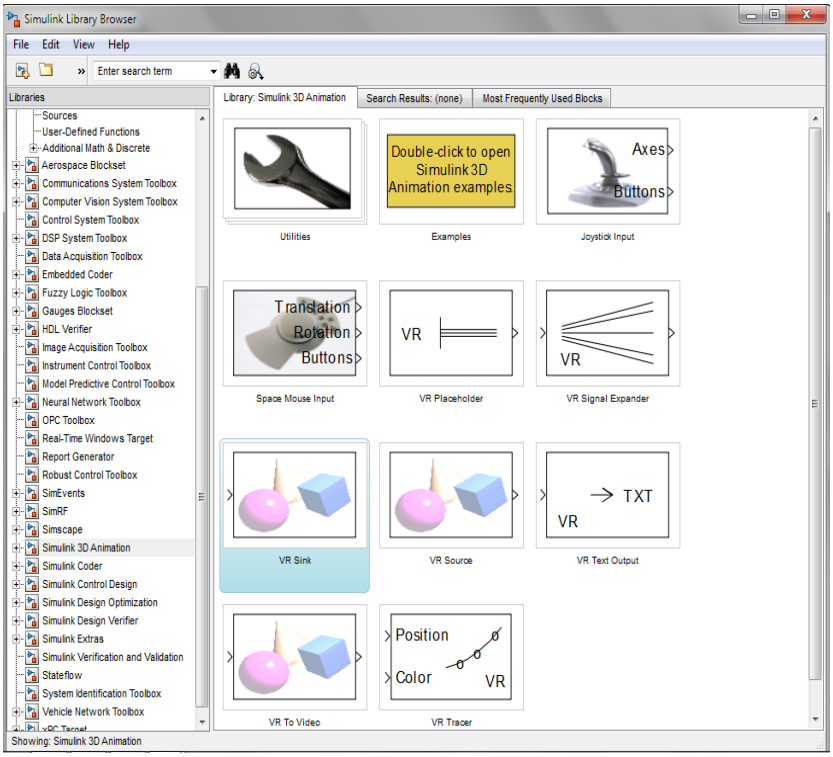


FIGURE 3.17 – *VR Sink dans la boite à outils de Simulink 3D Animation*

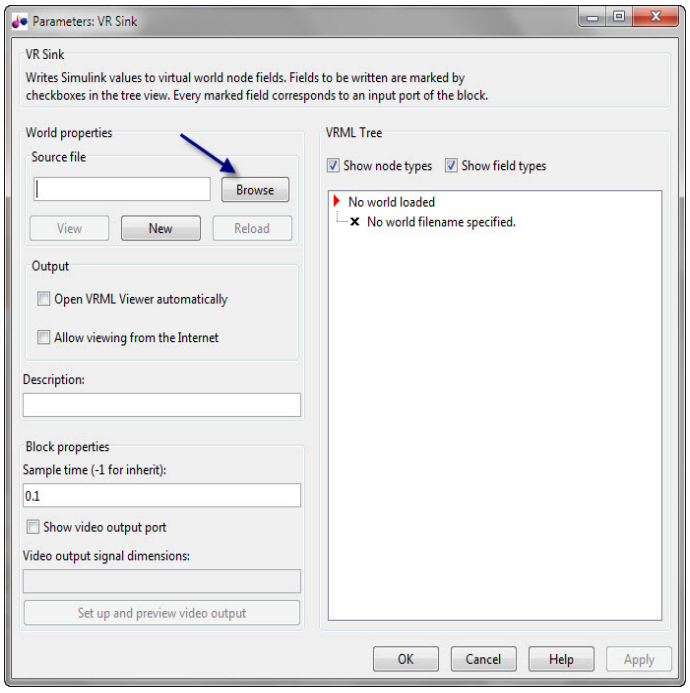


FIGURE 3.18 – *Bouton Browse pour charger la scène .wrl*

4. cliquer sur le signe(+) à la gauche du palette(Group) (*Fig.3.19*) ;

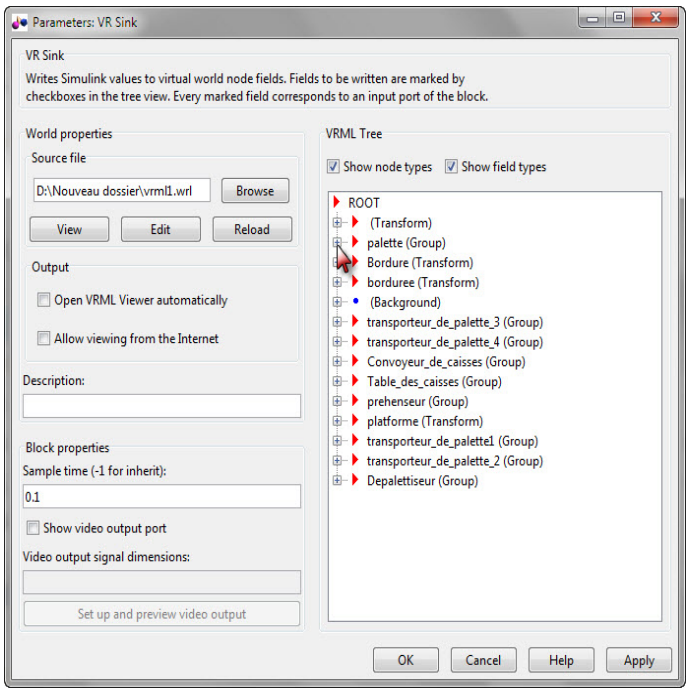


FIGURE 3.19 – *Palette Group*

5. cliquer sur le signe(+) à la gauche du palette(children) (*Fig.3.20*) ;

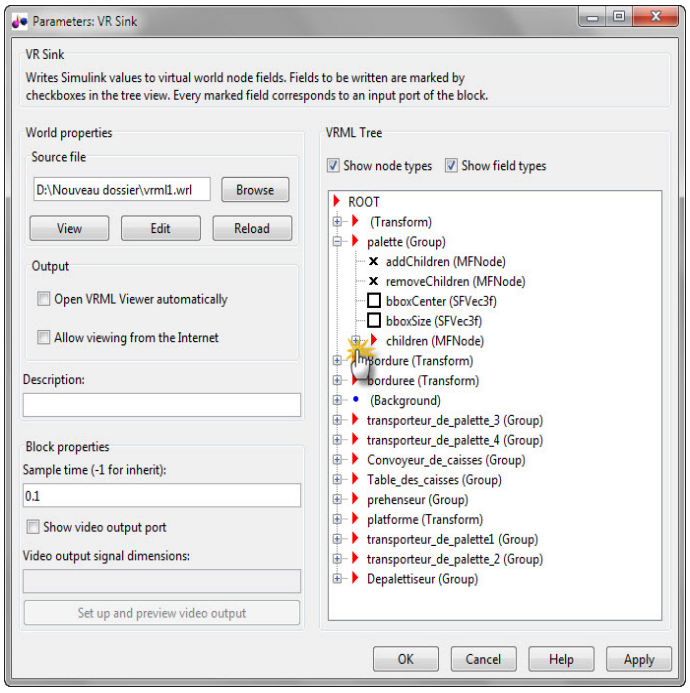


FIGURE 3.20 – *Palette children*

6. cocher sur le carré correspond à propriété de translation par exemple.(*Fig.3.21*), puis cliquer sur OK et quitter le VR.Sink ;

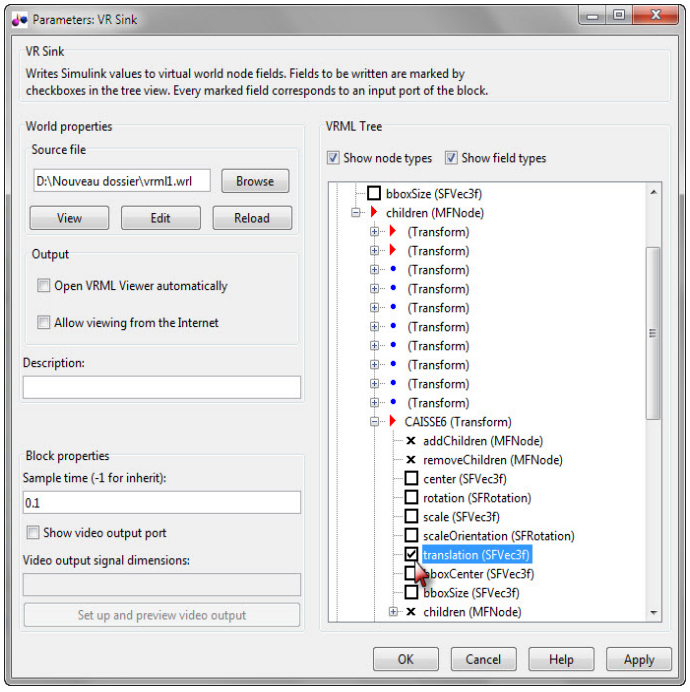


FIGURE 3.21 – *Propriétés de children(transform)*

Le schéma (Fig.8.7) montre au final le modèle qui est employé pour animer la scène virtuelle. on doit faire attention au sujet de la taille de dimension des signaux entrant dans le VR.Sink pour Pallette.translation. C'est un vecteur de 3×1 (Fig.3.22).

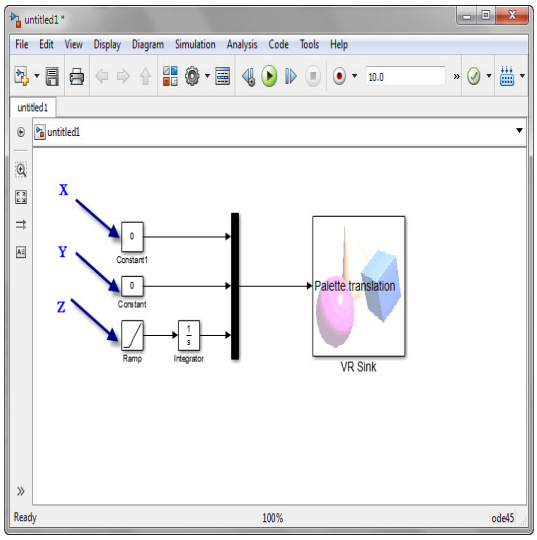


FIGURE 3.22 – *Modèle final*

3.4 Conclusion

La principale importance dans ce projet, est d'avoir développer toute l'interface en Simulink, depuis les blocs pour le contrôle de la chaîne de production, l'implémentation du modèle cinématique, dynamique et géométrique jusqu'à la visualisation dans le bloc de VRML de Simulink.

Avec cette simple application, on a fait une intégration complète de la simulation et de l'animation pour notre chaîne de production, dans le but d'explorer le vrai comportement du dépalettiseur avec le Transport de palettes dans un environnement virtuel.

4

Programmation avec Stateflow et Simulink

4.1 Introduction

Dans ce chapitre, on va procéder à la programmation du processus de fonctionnement de la chaîne de production sous Stateflow(Simulink/ MATLAB). Puis on passera à la simulation en temps réel à l'aide du logiciel 'Real-Time Windows Target' de MATLAB, suivit d'une présentation et interprétation des résultats de simulation.

4.2 Cahier des charges

c'est un document technique dans lequel sont définies l'ensembles des tâches à effectuer pendant une durée convenue et que l'on doit y soumettre afin de mener à bien toutes les opérations pré-définies, tout en respectant certaines closes.

4.2.1 Description du déroulement des différents systèmes

4.2.1.1 Transport de palettes

Les palettes sont introduites individuellement dans le sens transversal de la machine et elle sont détectées par un capteur de position (**cp**).

Dans ce cas, l'élément de transport est un segment de (*4*) convoyeurs à chaîne. La palette pleine se déplace jusqu'à l'emplacement de dépalettiseur(*station de palettisation*) où elle sera positionnée à l'aide des cellules photo-électriques(***cph1*** pour le deuxième convoyeur d'attente et ***cph2*** pour troisième convoyeur). La palette vide est soulevée à l'aide de désengageur d'auxiliaires d'emballage dans le magasin de stockage de palettes.

4.2.1.2 Dépalettiseur

En position initiale, l'élévateur se trouve au-dessus du troisième convoyeur et Le dispositif de centrage des couches reste ouvert. Ce dernier est indiqué par un capteur de proximité (***cp8***) et (***cp7***). Une fois la palette pleine entrée dans le troisième convoyeur, elle sera positionnée à l'aide d'un capteur photo-électrique cph2. L'élévateur s'abaisse jusqu'à ce que la cellule photo-électrique (***cph6***) soit actionnée, suivie d'une temporisation de 1 seconde. Dans cette position le dispositif de centrage des couches se ferme et encercle les articles qui se trouvent au dessous de la couche poussée par quatre (4) cloisons de centrages courtes. Les plaques de centrage latérales sont actionnées par un vérin pneumatique. Elles évitent un décalage et une chute d'articles. La sortie et l'entrée du vérin sont indiqués par un capteur de proximité ***(cp7)*** et ***(cp8)***, ensuite, le dispositif de l'entraînement de l'élévateur se déplace verticalement au-dessus de la palette jusqu'à l'activation du capteur qui indique le niveau maximum dans la colonne de l'élévateur.

La station de levage tourne à 90 degré au niveau de la table des couches à l'aide de deux capteurs inductifs qui indiquent l'angle de rotation qui sont : ***(cp2)*** pour la position initiale c-a-d 0 degrés et ***(cp3)*** pour la position de 90 degrés au dessus de la table des couches. Ensuite, L'entraînement de l'élévateur s'abaisse après avoir l'autorisation par un capteur photo-électrique (***cph4***) situé sur la table des couches et qui contrôle la disponibilité d'accueil pour les couches.

la station de levage descend jusqu'à la position de la table des couches à l'aide d'un capteur inductif (***cp5***) fixé sur la colonne de l'élévateur. une fois le vérin s'ouvre pour relâcher la couche, l'élévateur se remonte verticalement jusqu'au niveau maximum de la colonne de levage, dès lors, il retourne à la position initiale et il refait le même cycle

jusqu'à la dernière couche.

4.2.1.3 Table de couches

Après l'ouverture du dispositif de centrage, une libération de couche se produit,désormais il faudrait une temporisation d'une(*1*) seconde pour que le convoyeur de la table se mette en marche pour l'evacuation des caisses, indiquée par un capteur photo-électrique (***cph5***) qui, donne l'information si il y'a le embouteillage au niveau de la ligne d'évacuation des caisses ou non.

4.2.1.4 Evacuation des palettes

Les palettes vides sont soulevées par la station de levage dans le magasin de palettes.

Les palettes vides sont saisies par deux tôles de préhension après avoir quitté le convoyeur 4 indiqué par le capteur photo-électrique (***cph3***) et une temporisation de 5 secondes pour l'arrêt de ce convoyeur.

4.2.2 Définitions des entrées/sorties

Nous avons défini les entrées et les sorties du diagramme tout en essayant de s'approcher du système réel, et pour cela nous les avons défini comme suit :

4.2.2.1 Les entrées

Les entrées du système de transport de palettes :

On présente les entrées du transport de palettes(*capteurs*) avec leurs positions dans le tableau suivant :

Entrées	Type	Position sur la machine
cp	Capteur de position mécanique	Au milieu de 1^{ier} convoyeur
cph1	photo-électrique réflexe	Á la fin du 2^{eme}convoyeur
cph2	photo-électrique réflexe	Proche à la fin du 3^{eme} convoyeur
cph3	photo-électrique réflexe	Ála fin du 4^{eme} convoyeur

TABLE 4.1 – Les entrées du transport de palettes

Les entrées de système de dépalettiseur :

À ce stade, on présente les différents entrées de dépalettiseur (*capteurs*) avec leurs positions dans le tableau ci-dessous :

Entrées	Type	Position sur la machine
cp2	inductif	Sur le dispositif de rotation et en face du 3^{eme}convoyeur position 0°
cp3	inductif	Sur le dispositif de rotation et en face de la table de couches, position 90°
cp5	inductif	Fixé dans la colonne verticale au même niveau de la table de couches
cp6	inductif	fixé au niveau maximum du dépalettiseur
cp7	inductif	fixé sur la 1^{ere} extrémité du vérin qui indique sa sortie
cp8	inductif	fixé sur la 2^{eme} extrémité du vérin qui indique sa rentré
cph6	Ph-électrique	fixé dans l'élévateur pour détecter le niveau de la couche

TABLE 4.2 – Les entrées de système de dépalettiseur

Les entrées de système de la table de couches :

Finalement on présente les différents entrées de la table de couches(*capteurs*) avec leurs positions dans le tableau ci-après :

Entrées	Type	Position sur la machine
cph4	photo-électrique	Á l'extrémité du milieu de la table de couches.
cph5	photo-électrique	Á la fin de l'extrémité de la table et en face du convoyeur de caisses.
cph7	photo-électrique	Dans la barrière de sécurité en face du dépalettiseur.
Bp	Bouton poussoir	au niveau de la pupitre de commande pour la mise en marche ou arret.
Bur	Bouton poussoir	au niveaux de la barrière de sécurité et pupitre de commande.

TABLE 4.3 – Les entrées de la table de couches

4.2.2.2 Les sorties

Les signaux de sorties sont visualisés dans un scope qui indique l'état de marche et d'arrêt des différents organes du procédé :

Moteur qui entraîne le 1^{ier} convoyeur : Mis en Marche$\Longrightarrow$ le signal devient un 1, Mis en arrêt$\Longrightarrow$ le signal de sortie reçoit un 0

Moteur qui entraîne le 2^{eme} convoyeur : 1$\Longrightarrow$ Marche, 0$\Longrightarrow$ Arrêt

Moteur qui entraîne le 3^{eme} convoyeur : 1$\Longrightarrow$Marche, 0$\Longrightarrow$ Arrêt

Moteur qui entraîne le 4^{eme} convoyeur : 1$\Longrightarrow$Marche, 0$\Longrightarrow$ Arrêt

Moteur d'entraînement de l'élévateur : 1$\Longrightarrow$Marche, 0$\Longrightarrow$ Arrêt

Moteur de rotation du dépalettiseur de 0° à 90° 1$\Longrightarrow$Marche, 0$\Longrightarrow$ Arrêt

Vérin à double effet ouvert au repos : lorsque cp7=1$\Longrightarrow$ vérin sortie =1 et lorsque cp8 =1$\Longrightarrow$ vérin entré =1 et vis versa

Moteur d'entraînement de la table de couches : 1$\Longrightarrow$Marche, 0$\Longrightarrow$ Arrêt

Pour mieux expliquer, on a indiqué les différents capteurs et actionneurs sur la figure(*Fig. 5.5*) : (voire l'annexe)

4.2.2.3 Les Temporisations

Dans le système étudié on a trois temporisations :

Temporisation au niveau de transport de palettes : Elle se déclenche pendant 5 seconde après l'arrêt du 3^{eme} convoyeur afin d'arrêter le 4^{eme} convoyeur ;

Temporisation au niveau de dépalettiseur : elle commence dès que le (***cph6=1***) afin de centrer le niveau de couche et ça dure 1 seconde ;

temporisation au niveau de la table de couche : elle démarre lorsque le dispositif de prehension relâche la couche, afin d'éviter la collision des couches, c-à-d dès que (***cp7=1***) la temporisation déclenche de 1 secondes, après le moteur de la table peut fonctionner selon l'état de (***cph5***).

4.3 Programmation de processus de fonctionnement

4.3.1 Construction d'un diagramme

Avant de représenter le diagramme il faut procéder à l'analyse du problème à traiter ; dans notre cas, il s'agit de simuler le modèle d'une partie d'un système de production. De cet examen doivent se dégager peu à peu les différents états qui correspondent aux fonctions de système. Puis chaque fonction sera détaillée à son tour. Et les états correspondant seront mis à jour, et ainsi jusqu'au stade ultime où tous les aspects auront été pris en compte par consequent, toutes les actions seront repérées et toutes les conditions seront répertoriées. Cette construction de diagramme est une opération très délicate de modélisation mais dont la pertinence est essentielle.

La descendance d'un parent doit être, soit du type **'ou exclusif'**(*or*) ou bien du type **'parallèle'** (*and*). C'est par leurs bordure qu'on distingue les deux types, l'état **'ou exclusif'** a une bordure pleine et celle de l'état **'parallèle'** est en trait pointillé. Lorsqu'un parent est actif tous ces enfants le seront si la décomposition est de type 'OR'.

Il appartient au concepteur de signaler ce seul état actif au moyen d'une transition particulière appelée, en anglais, **defaut transition**

4.3.2 Diagramme Stateflow de transport de palettes

Après une analyse fonctionnelle du processus, et à l'aide de la table de vérité on a déduit les différents cas possibles de fonctionnement de transporteur de palettes. Étant donnés que les palettes arrivent dans la ligne individuellement, donc notre système doit agir en parallèle et a tout moment pour gérer les déplacements des palettes jusqu'au magasin de stockage pour les palettes vides.

La table de vérité nous permet de définir les quatre cas possibles de fonctionnement :

cp	cph1	cph2
1	0	0
1	1	0
1	0	1
1	1	1

TABLE 4.4 – La table de vérité

Initialement, la palette arrive au premier convoyeur sachant que le deuxième et le troisième sont libres, et on le considère comme le premier cas de fonctionnement. Après, les autres palettes qui arrivent seront traitées dans les quatres cas suivant :

1er **cas :** l'arrivée d'une nouvelle palette viens juste après l'évacuation de la précédente palette vide dans le deuxième convoyeur *(cp=1, cph1=0 et cph2=0)*

2eme **cas :** l'arrivée d'une nouvelle palette sachant que la précédente soit dans le deuxième convoyeur *(cp=1, cph1=1 et cph2=0)*

3eme **cas :** l'arrivée d'une nouvelle palette sachant que la précédente se trouve dans le troisième convoyeur *(cp=1, cph1=0 et cph2=1)*

4eme **cas :** l'arrivée d'une nouvelle palette sachant que les deux convoyeurs sont occupés, *(cp=1, cph1=1 et cph2=1)*

Pour cette application, nous avons présenté les différents diagrammes de statecharts qui traitent en parallèle l'arrivé des palettes :

Nous trouvons dans cette figure les états du système avec moins d'arc et plus de lisibilité pour savoir à tout instant la situation dans laquelle se trouve notre système.

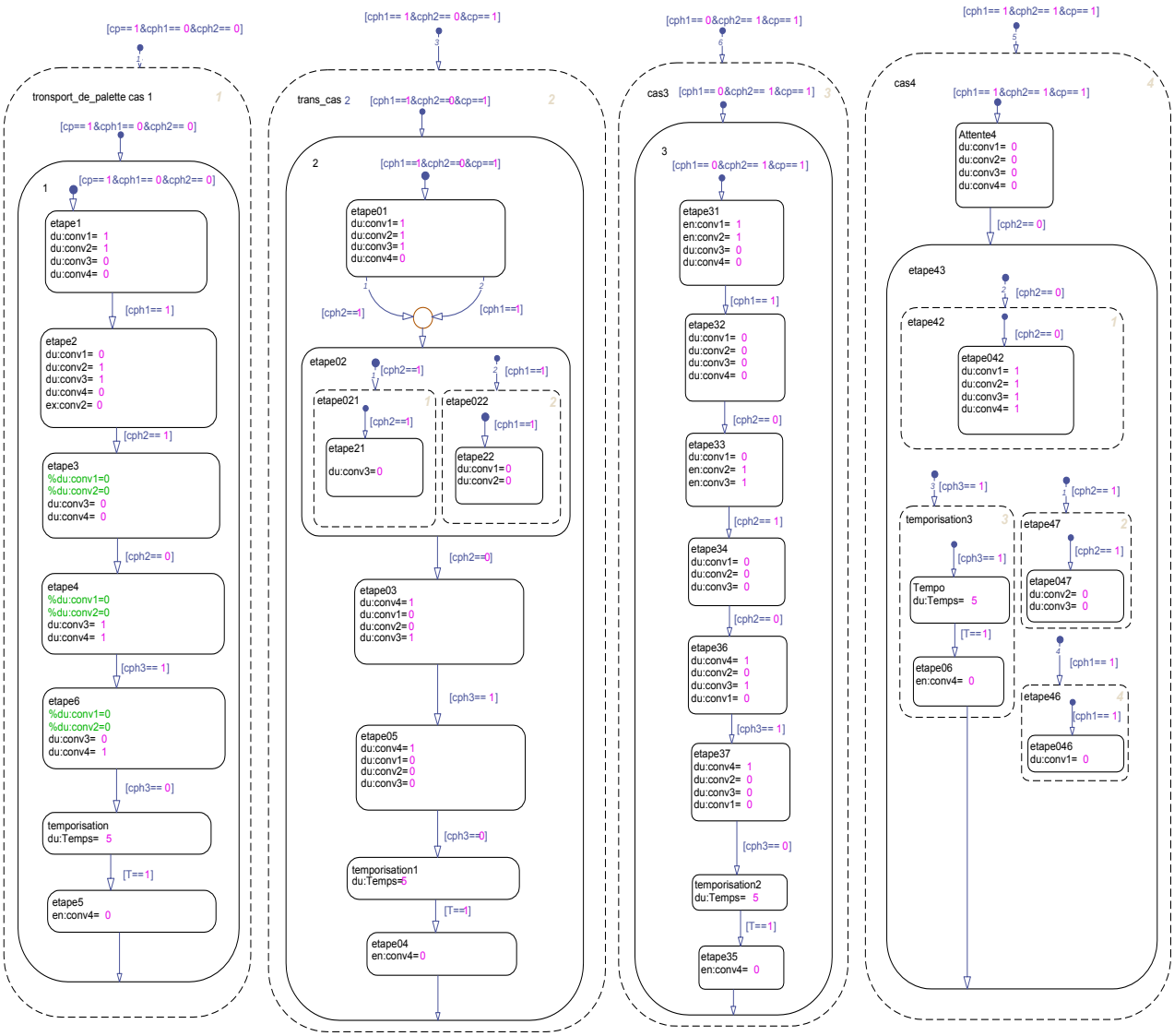


FIGURE 4.1 – *Diagramme de transport de palettes*

4.3.3 Diagramme Stateflow du dépalettiseur

Dès que le capteur (***cph2***) marque la présence d'une palette au niveau du troisième convoyeur *(**cph2=1**)*, le processus de dépalettiseur démarre l'evacuation des couches. Et voici le diagramme stateflow qui traite l'ensemble des séquences :

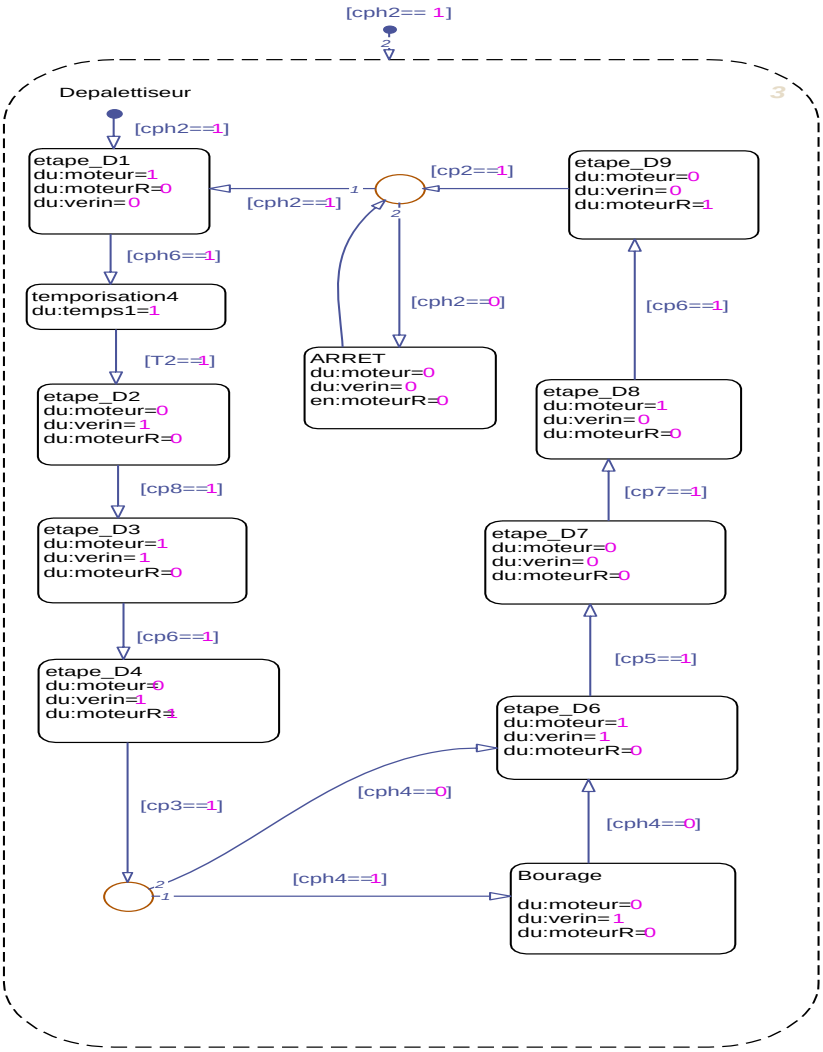


FIGURE 4.2 – *Diagramme de dépalettiseur*

4.3.4 Diagramme Stateflow de la table des caisses

Dès que le dispositif de centrage dépose la couche sur la table (***cp7=1***), le processus de la table démarre après une temporisation d'une seconde. Sachant que les trois systèmes (transport de palette, dépalettiseur et la table des couches) fonctionnent en parallèles, le diagramme de stateflow ci-dessous explique les étapes de fonctionnement.

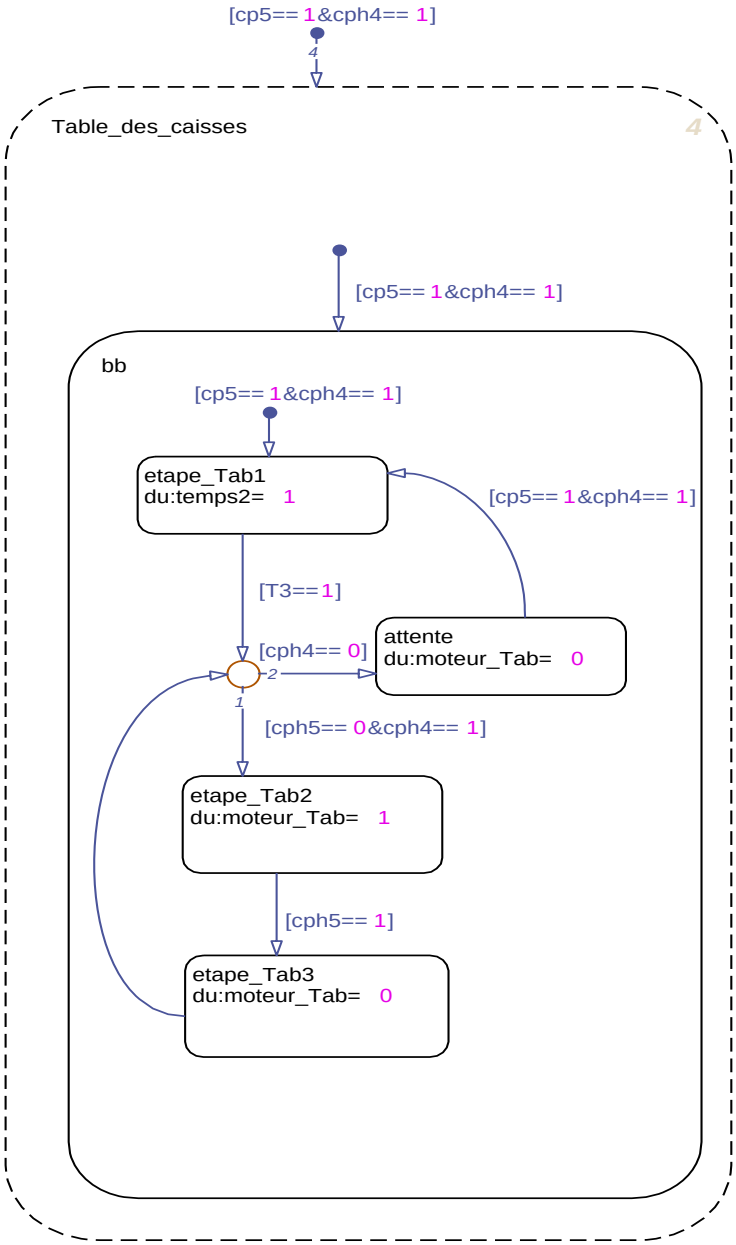


FIGURE 4.3 – *Diagramme de table de couches*

4.4 Programmation

A base de ce que nous avons vu et défini dans le cahier des charges comme paramètres, nous avons programmé le processus de fonctionnement de la partie de la chaîne de production sous Stateflow(Simulink).

4.5 Schéma bloc du système

Le schéma bloc hybride est représenté sur la figure suivante :

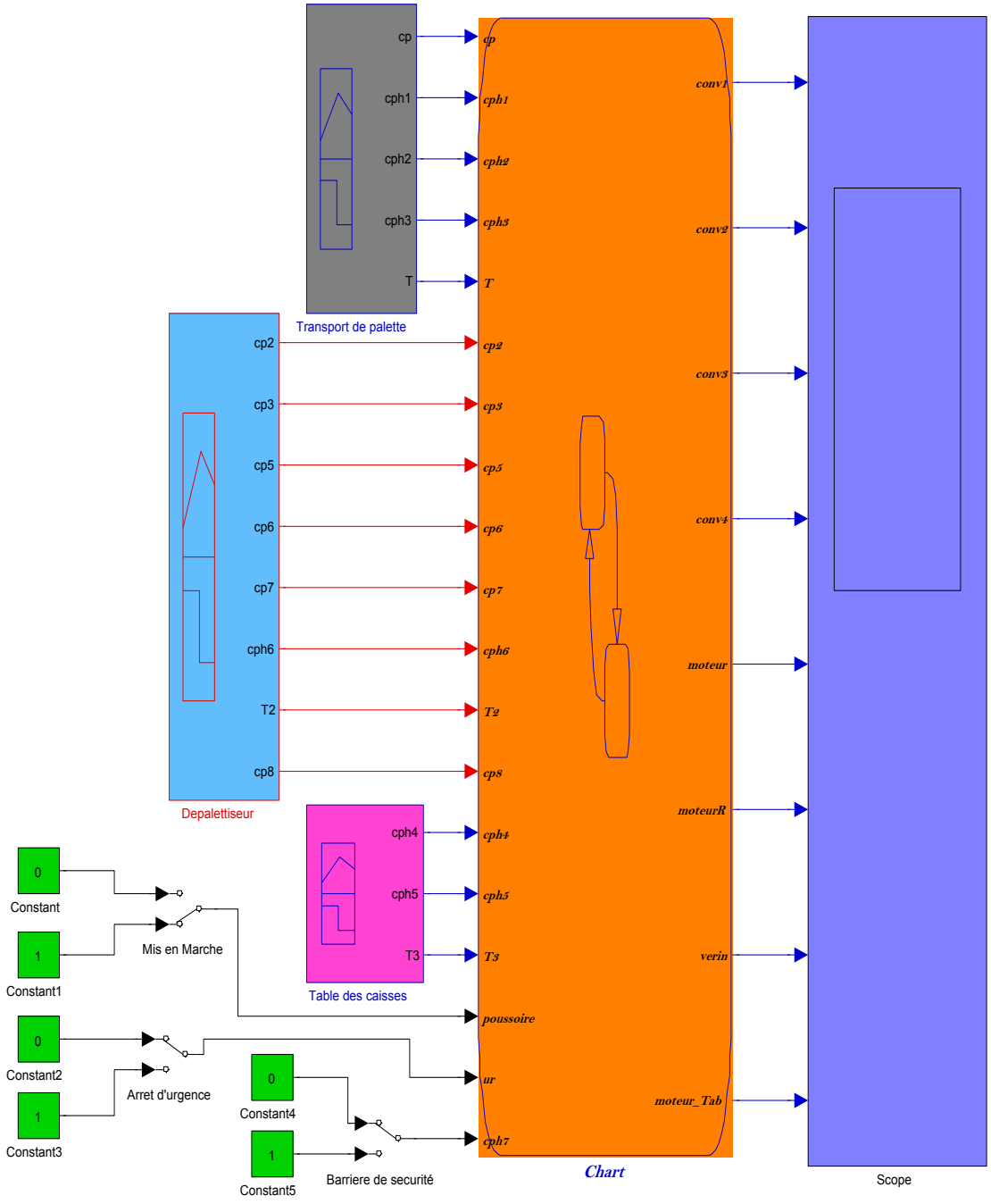


FIGURE 4.4 – *Schéma bloc de simulation*

Nous avons à l'entrée du chart les différents signaux d'entrée, présentés dans les blocs (signal builder) et trois switch : un pour la mise en marche ou arrêt, l'autre pour l'arrêt

d'urgence et le troisième switch pour la barrière de sécurité qui arrête le processus en cas l'affranchissement de cette barrière.

Á la sortie du chart, nous avons les différents signaux de sortie*(différents actions)* qui sont visualisés sur un scope simple.

4.6 Diagramme Stateflow du programme global

Cette application diffère des précédentes du fait qu'elle contient plusieurs activités qui s'exécutent en parallèle, les traitement des palettes ainsi que les déplacements du dépalettiseur entre les différents postes de traitement. Nous remarquons dans le diagramme représenté dans la figure(*Fig. 5.7*) (voire l'annexe) que les tâches de chacune des activités sont actives simultanément, à savoir les durées de traitement et les durées de déplacement du robot entre les différents postes (transport de palette, magasin de palette et la table de couches).

4.7 Schéma bloc du système avec v-realm builder

Dans le cas de la simulation avec Simulink en 3D, les signaux de sortie vont attaquer directement le VR.Sink de V-realm builder, sachant que les entrées du bloc VR.Sink sont les différents objets orientés (translation, rotation ...etc), et pour synchroniser le processus de fonctionnement, on a ajouté des signaux qui gèrent la scène en 3D et on a utilisé la boite à outils (bibliothèque) de Simulink.

Le schéma bloc de Simulink est présenté sur la figure(*Fig 5.6*) (voir l'annexe).

4.8 Simulation en temps réel

4.8.1 Configuration

Pour simuler notre modèle en temps réel, on utilise le logiciel de MATLAB 'Real-Time Windows Target'[10], pour cela on suit les étapes suivantes :

1. On crée un répertoire de travail ;
2. On installe le logiciel 'Real-Time Windows Target' dans l'ordinateur. Cette installation se fait on introduisant cette commande dans la fenêtre de commande de MATLAB : rtwintgt -install puis on tape 'yes' ;

3. On configure notre modèle en tapant cette commande dans la fenêtre de commande de MATLAB : rtwinconfigset('nom du fichier simulink'), et dans notre modèle par : rtwinconfigset('programmeglobal') ;
4. Dans le modèle de Simulink, dans le menu 'Simulation' on sélectionne 'Configuration Paramètres'. Une fenêtre apparaît ;
5. On sélectionne le volet 'Solver', et dans 'Type' on sélectionne l'option 'Fixed Step' ;
6. Dans 'Fixed step size' on introduit la période d'échantillonnage 'Te' ;
7. On clique sur 'Apply' pour enregistrer les modifications ;
8. Dans le menu 'Tools' on sélectionne 'Real-Time Workshop' puis on sélectionne 'Build model'. Après cette étape MATLAB crée Le code exécutable ;
9. Après la création du code exécutable, on clique sur l'icône 'Connect to Target' ;
10. Et enfin on lance la simulation en clinquant sur l'icône 'Start', ou dans le menu 'Simulation' on sélectionne 'Start Real Time Code'.

4.8.2 Résultats de Simulation

On a simulé l'entré d'une palette à 5 couches qui traite le premier cas, après quelques minutes une autre palette rentre dans la ligne qui correspond au troisième cas.

Après avoir simulé notre programme, nous avons présenté les résultats de simulation sur les figure suivantes, qui sont les signaux de d'entrée et les signaux sortie .

Les signaux présentés sur les figures ci-dessous correspondent aux informations de type TOR (1 ou 0) des différents capteurs et actions du système global :

4.8.3 Les signaux d'entrée

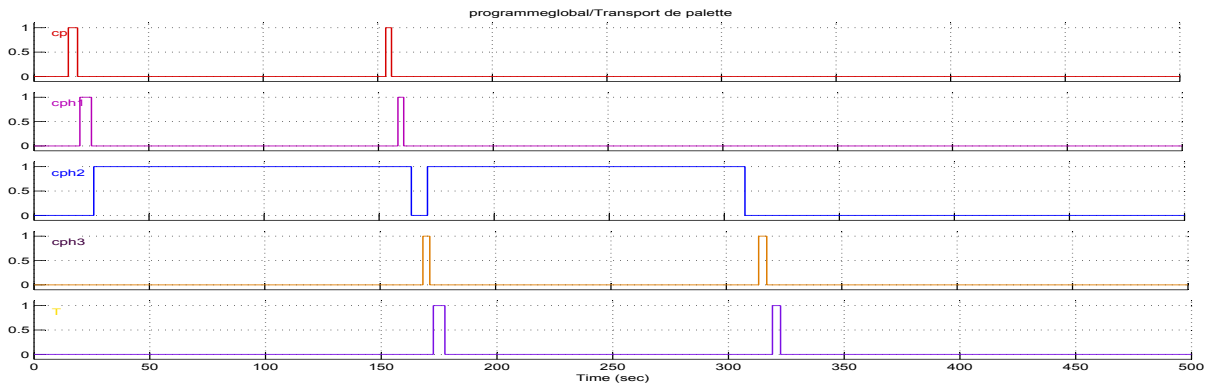


FIGURE 4.5 – *Les signaux d'entrée de transport de palettes*

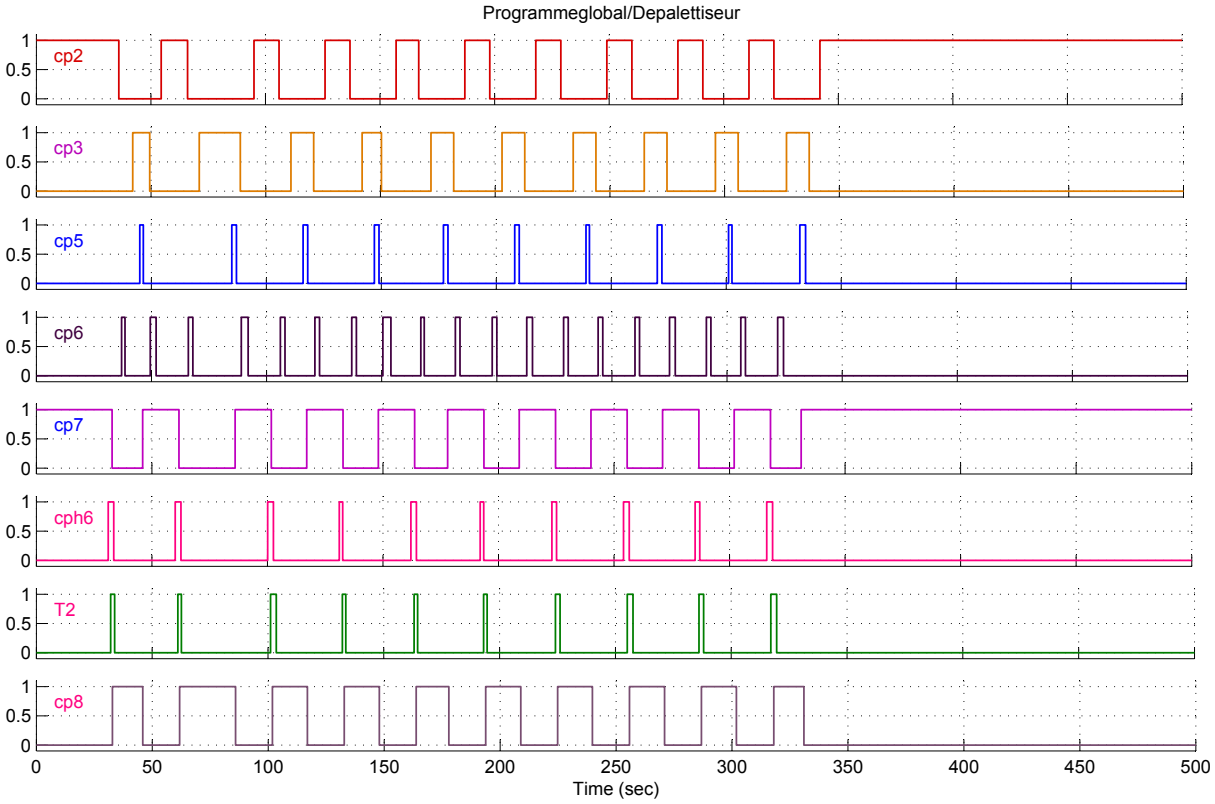


FIGURE 4.6 – *Les signaux d'entrée de dépalettiseur*

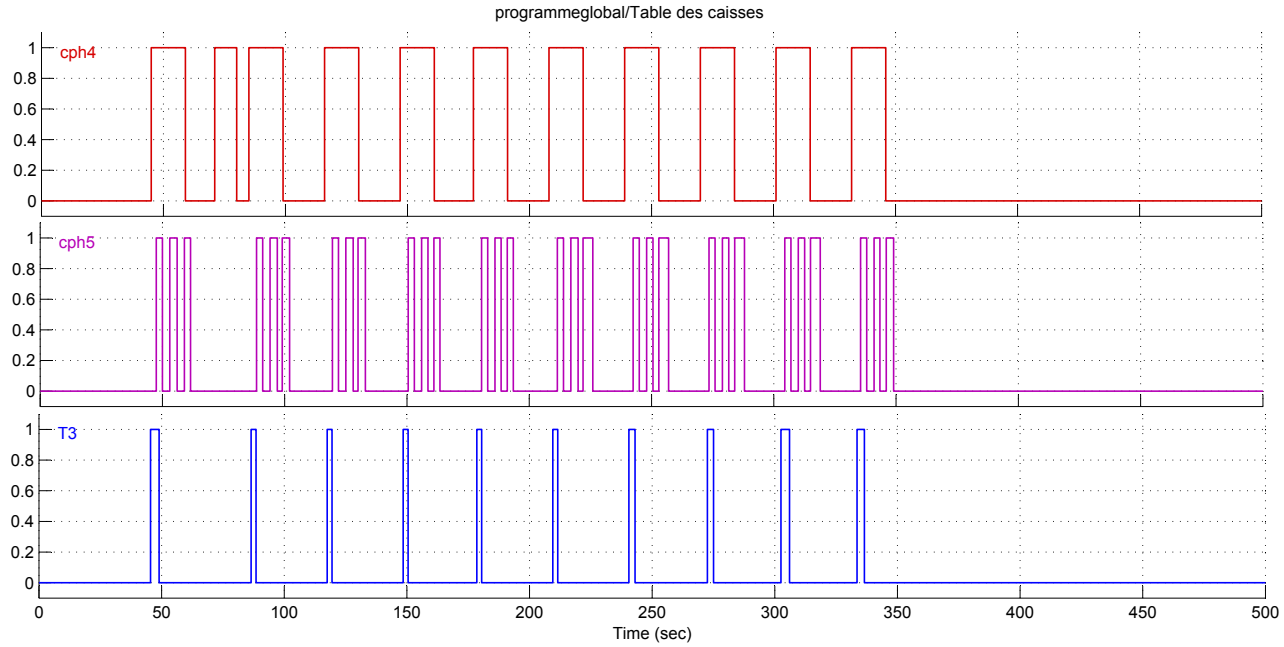


FIGURE 4.7 – *Les signaux d'entrée de la table de couches*

4.8.3.1 Les signaux de sortie

Après l'arrêt de la simulation on a importé les signaux de sortie (actions) qui explique le processus de fonctionnement et qui correspond bien au cahier des charges et qui respecte la table de vérité qu'on a définit auparavant.

La figure suivante donne les résultats de la simulation en temps réel :

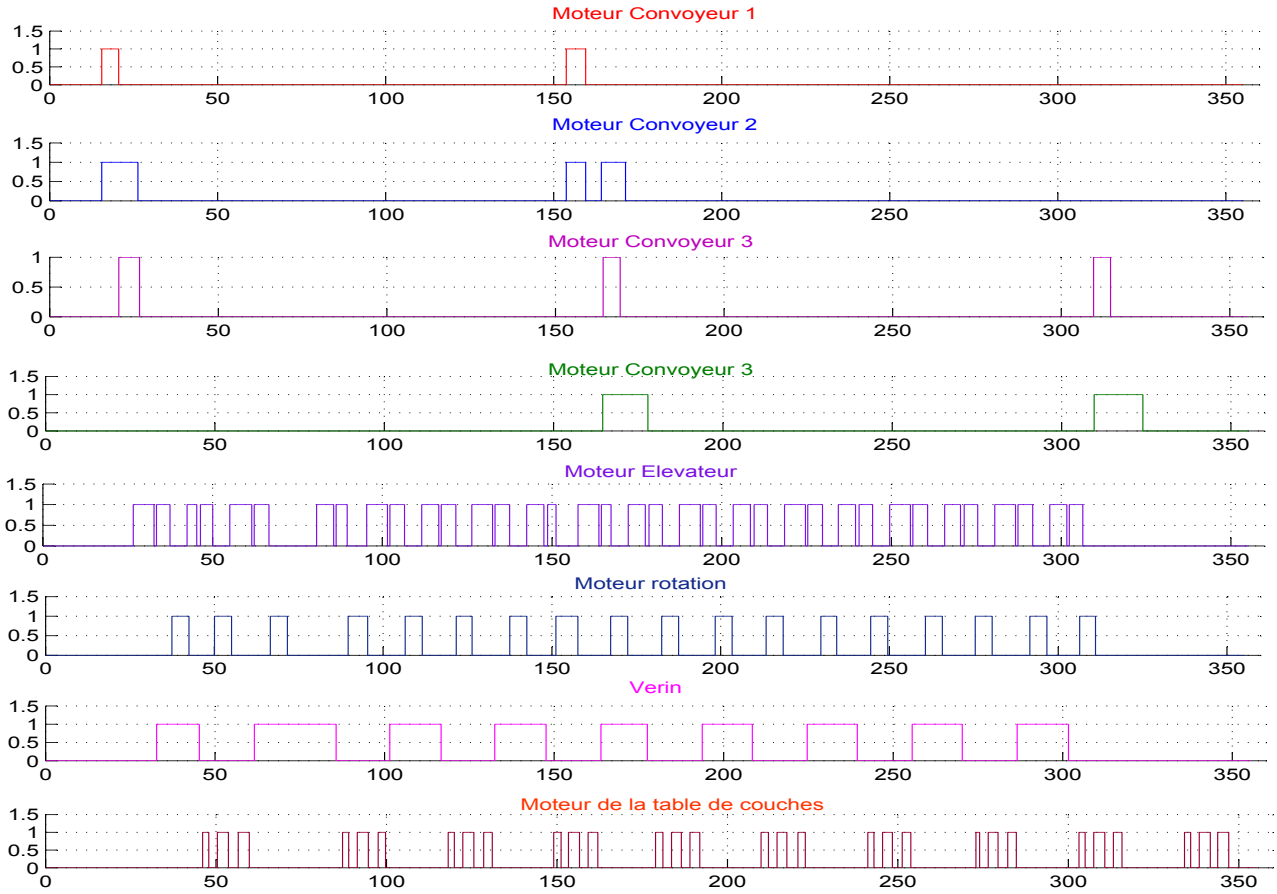


FIGURE 4.8 – *Résultats de simulation*

4.9 Conclusion

Au cours de ce chapitre, on a présenté notre programme de simulation avec et sans l'environnement 3D qu'on a ensuite simulé en temps réel sous Simulink de MATLAB à l'aide du logiciel 'Real-Time Windows Target' de MATLAB. Puis on a présenté les résultats de simulation. Ces résultats montrent que le programme qu'on a dressé satisfasse amplement le cahier des charges.

5

Implémentation du programme de commande sur la carte dSPACE

5.1 Introduction

Ce chapitre est consacré à l'implémentation du programme réalisé au chapitre précédent dans la carte dSPACE ds1104 pour une autre simulation en temps réel.

La nouvelle génération des DSP de dSPACE est composée des dispositifs les plus compactes qui existent à ce jour et qui sont dédiés au développement des prototypes de contrôle rapides. Ils réduisent le temps et le coût de développement des nouveaux algorithmes de contrôle. Leurs avantages principaux sont la simplicité d'utilisation et la commodité. En effet, l'interfaçage direct avec Simulink/MATLAB permet de concevoir les algorithmes de contrôle utilisant les blocs diagrammes graphiques de Simulink pour ensuite les faire valider directement en temps réel [11].

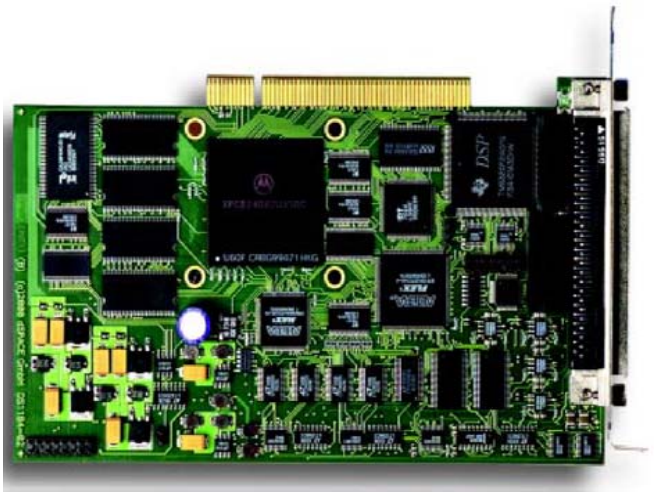


FIGURE 5.1 – *La cart Dspace*

Particulièrement, le kit ACE de dSPACE est un outil très attrayant destiné pour les applications universitaires. Il existe deux versions différentes du kit ACE de dSPACE :

- le ACE kit 1104
- le ACE kit 1103

Les avantages de ces kits sont nombreux, puisqu'ils offrent la possibilité de :

- Tester les méthodes de contrôle les plus complexes en temps réelles ;
- Se concentrer uniquement sur la conception de la technique de commande désirée ;
- Optimiser en temps réel les contrôleurs ;
- Développer l'expérience pratique en milieu universitaire ;
- Travailler avec une interface Windows facile à utiliser ;
- Implanter les modèles faits dans Simulink directement et les exécuter en temps réel dans dSpace ;
- Observer le comportement des systèmes lors de changements des paramètres de façon automatique.

5.2 Domaine d'application

Les champs d'application typiques de dSpace sont :

- Les drives et les moteurs ;
- Les moteurs à inductions triphasés ;
- Le contrôle des bruits et vibrations ;

– L'hydraulique et la pneumatique ;
– La robotique ;
– l'automatique et traitement de signal.

La facilité de l'utilisation de ce kit, provient de l'interfaçage direct avec Simulink. En effet, une fois la méthode de contrôle développée et conçue dans Simulink, soit par des blocs diagrammes soit par des programmes en langage C, elle est compilée. Ensuite les données sont envoyées à la carte dSpace pour que l'application démarre en temps. Le schéma suivant illustre toutes ces étapes :

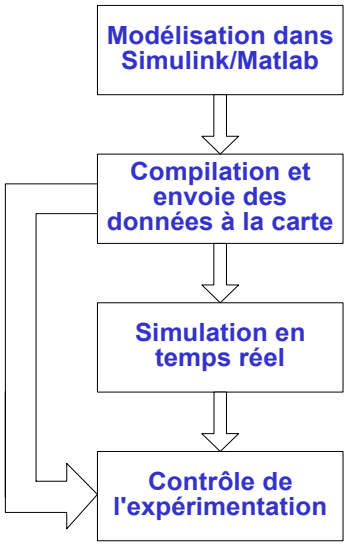


FIGURE 5.2 – *Étape d'utilisation de Ds1104*

5.3 Real Time Interface (RTI)

La programmation d'un modèle, nécessite un logiciel qui effectue le lien entre la carte dSPACE et le logiciel Simulink de MATLAB. La carte dSPACE est munie d'une interface en temps réel (RTI) qui permet d'établir un lien avec Simulink afin de lui ajouter des interfaces I/O, des interruptions, et même d'établir le code en temps réel qui sera téléchargé et exécuté sur la dSPACE.

Avec la Real-Time Interface (RTI), on peut facilement faire fonctionner des modèles de fonction sur la carte ds1104, configurer toutes les E/S graphiquement, insérer les blocs dans un schéma-bloc Simulink et générer le code du modèle via le Simulink Coder (Real-Time Workshop). Le modèle temps réel est alors compilé, téléchargé et démarré automa-

tiquement ce qui réduit le temps d'implémentation.

5.4 Démarrage de ControlDesk

Le logiciel ControlDesk est une interface graphique, qui permet l'administration de la carte dSPACE ainsi que la gestion des applications et les fonctions de contrôles. Ce logiciel s'adapte à n'importe qu'elle carte dSPACE dsxxxx.

1. Dans le menu start, choisir dSpaceTools-ControlDesk
2. Dans la fenêtre du ControlDesk, cliquer sur Platform

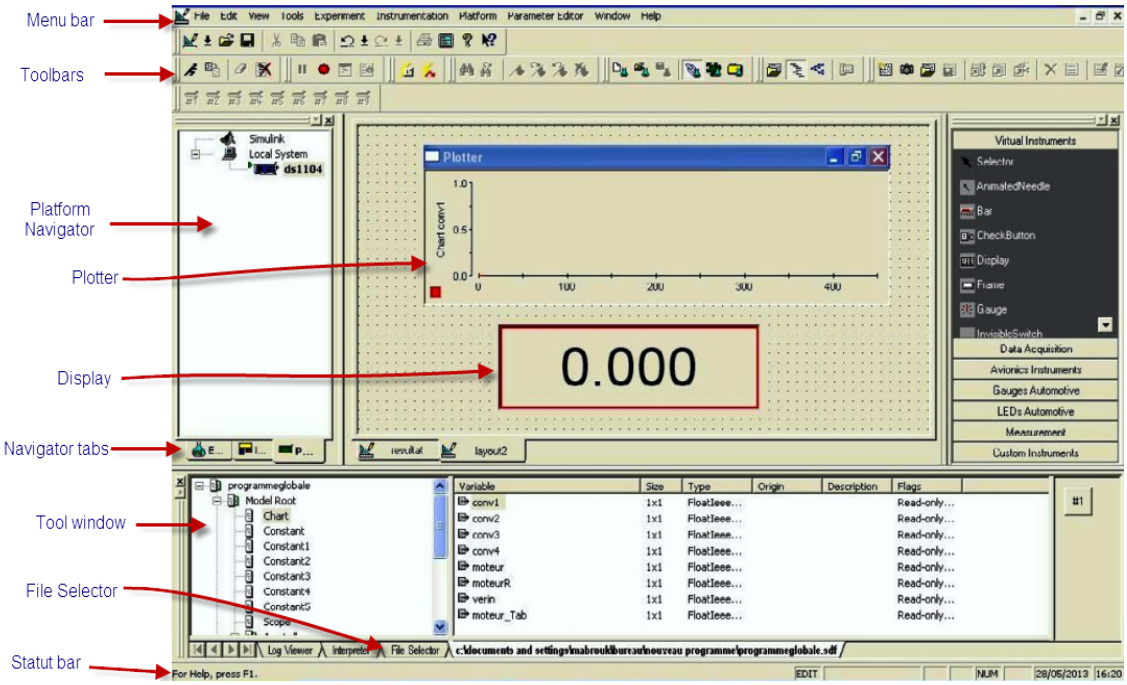


FIGURE 5.3 – *Fenêtre principale de ControlDesk*

5.4.1 Chargement d'un modèle Simulink dans ControlDesk

Pour charger un modèle Simulink après la compilation, on dispose de plusieurs méthodes, on peut choisir les étapes suivantes :

- Glisser et déposer le fichier d'extension (SDF) à la platforme Simulink dans le navigateur de plate-forme ;
- Cliquer sur le bouton 'Load Application/Model' dans la barre d'outils (Toolbar) de la 'platforme manager' et sélectionner le fichier à charger ;

– Ouvrir le menu contexte d'une variable (variable tree) et cliquer sur 'assign platforme' pour lancer MATLAB et charger le modèle associé.

5.4.2 Verification de l'installation

– Pour vérifier le bon fonctionnement du DS1104, il faut basculer de ≪ file selector ≫ à ≪ log Viewer≫ et si aucune erreur n'est rapportée, la carte opère adéquatement
– Si l'application est adéquatement chargée, les variables mises en jeu par l'application sont disponibles sous le répertoire Model Root
– La visualisation des variables nécessite d'ouvrir un nouveau plan d'affichage en choisissant dans le menu New Layout et ensuite choisir l'instrument nécessaire à la visualisation, par exemple choisir un traceur (plotter) dont les propriétés peuvent être personnalisées à tout moment

5.5 Implémentation du programme dans la carte Ds1104

Áfin d'implémenter un programme dans la carte Dspace on doit suivre les étapes suivantes :

1. Premièrement, on choisit la carte Dspace qu'on utilise ;
2. une fois la carte est installée, on accède au modèle Simulink ;
3. Puis on compile le modèle avec RTI (Real-time Interface).

Une fois la compilation est terminée on charge le fichier d'extension (SDF) dans la RTI1104 dans la fenêtre du ControlDesk, puis on ouvre un nouveau **layout** et à travers l'**Edit mode** on trace les différents plotter de notre sortie de chart, ainsi un bloc de **capture frame** pour enregistrer l'animation en temps réel. Après, on passe au mode animation.

Á la fin de l'animation, on sauvegarde les donnés de la capture dans un fichier d'extension(.mat), on charge le fichier dans MATLAB pour visualiser les résultats de la simulation en temps réel, et pour cela on a suivi les étape suivante :

data=load('resultatreel')

data.resultatreel

ans =

X : [1x1 struct]
Y : [1x8 struct]
Description : [1x1 struct]
RTProgram : [1x1 struct]
Capture : [1x1 struct]

T=data.resultatreel.X

T =

Name : ”
Type : 4
Data : [1x1198 double]
Unit : ”

T=data.resultatreel.X.Data ;
Y=data.resultatreel.Y(1).Data ;
Et finalement on tape :

plot(T,Y)

pour voir le premier graphe concernant le moteur du convoyeur 1.

Remarque : on suit les mêmes étapes pour afficher les autres graphes concernant les différentes sorties de notre modèle.

La figure ci-après montre les résultats de l'implémentation dans la carte Dspace1104 :

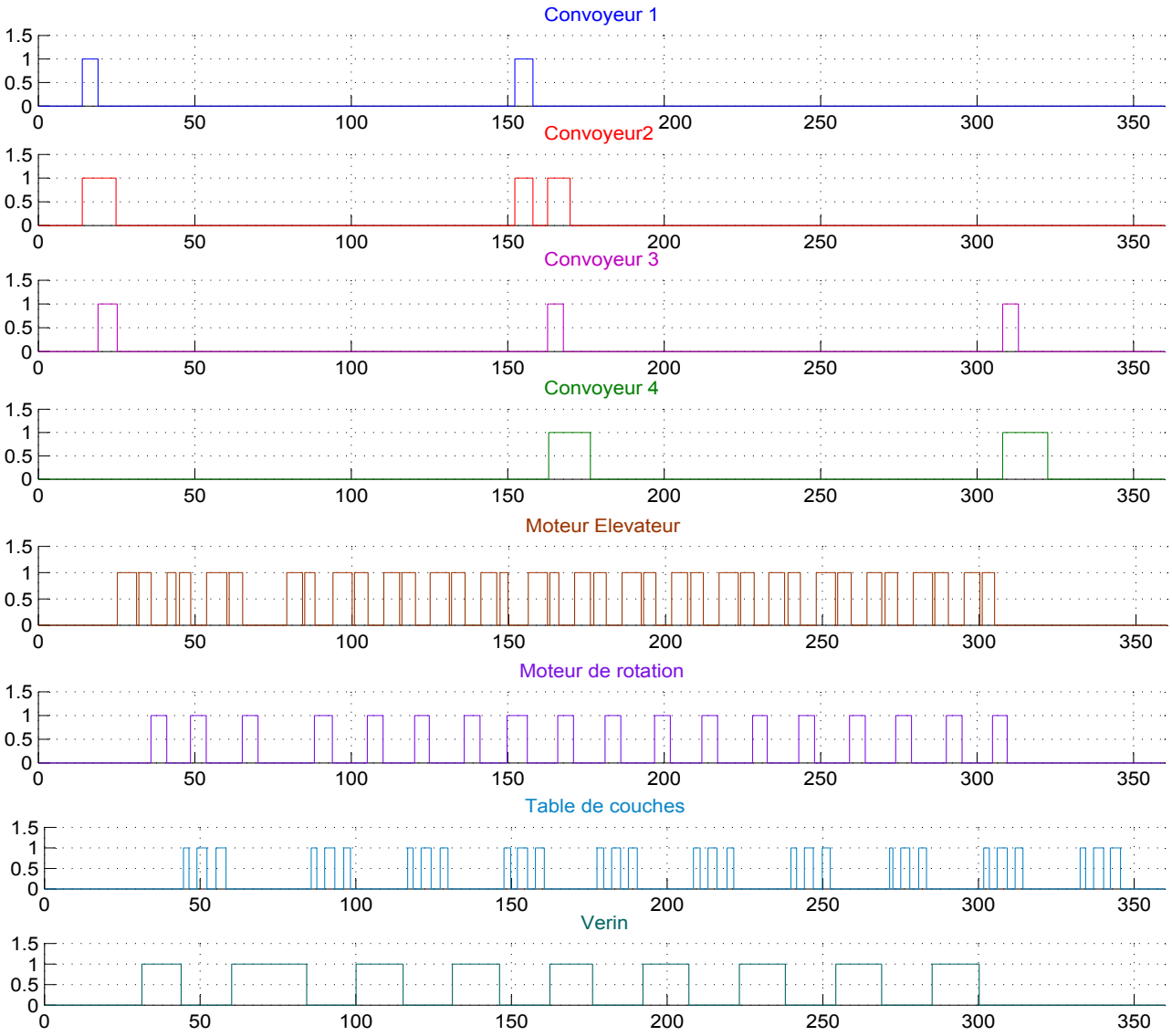


FIGURE 5.4 – *Les signaux de sortie avec RTI1104*

5.6 Conclusion

Dans ce chapitre nous avons procédé à l'implémentation et la validation de la simulation en temps réel du programme dans la carte dSPACE ds1104.

Conclusion générale

L'objectif du travail présenté dans ce mémoire est de constituer une nouvelle approche de l'automatisation des systèmes à événements discrets. En se basant sur deux toolbox puissants de MATLAB. En l'occurrence, le Stateflow et le V-Realm Builder.

Notre démarche consiste à définir les objets constituant le système étudié d'une part, et décrire l'ensemble des tâches de différents organes . Ensuite, à partir de l'outil Simulink 3D(V-Realm Builder) nous avons crée un modèle Simulink pour animer la scène virtuelle(la partie opérative), pour établir le modèle de commande de cette dernière, nous avons fait appel à un outil structurel adapté à la description et la commande des systèmes à événement discret qui est le 'Stateflow' .

L'environnement Simulink nous a permis de créer un modèle hybride(association de Stateflow et V-Realm Builder dans Simulink), lequel pourrait simuler le processus entier.

Nous avons implémenté le modèle de commande en utilisant la carte dSPACE (dS1104) afin de tester le programme en temps réel.

L'application Simulink/Stateflow et V-Realm Builder sous MATLAB pour l'automatisation et la simulation en 3D du système de production a été testée, en dépit de certaines complexités qui ont rendu la tâche très ardue :

Lors de notre projet le programme est devenu volumineux et encombrant, mais nous pensons qu'on pourrait réduire sa taille afin qu'il soit lisible et extensible, en exploitant mieux les fonctionnalités du Simulink/Stateflow. Par ailleurs, nous avons aussi rencontré des difficultés pour simuler la génération des signaux de capteurs dont il reste à chercher la solution.

Le travail présenté ici est bien sûr reste à parfaire et à affiner, de nombreux tests et réflexions qui auraient pu être effectuées. Ils sont mentionnées ici comme pistes à explorer :

– Il aurait été souhaitable d'améliorer le modèle de programmation, en exploitant mieux les outils Stateflow et V-Realm Builder

- Il convient de mettre en oeuvre un programme qui permet de générer l'état logique des détecteurs(capteurs) pour la simulation ;
- L'étude de l'aspect sécurité et fonction de diagnostic, qui a été complètement occulté au cours de ce travail, mériterait d'être entreprise.

Enfin, ce modeste travail était beaucoup plus un apprentissage dans la pratique de l'automatisme, une discipline qui ne cesse d'évoluer avec le développement des équipements technologiques et informatiques.

Bibliographie

[1] Z. De-feng, Z.Shu-hai, and L. Guo-xi. Matlab graph and animation design. *Beijing*, 2009.

[2] Steven T. Karris. *Introduction to Stateflow with Applications*. Orchard, 2007.

[3] Nassim Khaled. *Virtual Reality andAnimation for MATLAB and Simulink Users*. Springer, 2012.

[4] N. Martaj and M. Mokhtari. *MATLAB R2009, Simulink et Stateflow pour Ingénieurs, Chercheurs et Etudiants*. Springer, 2010.

[5] The MathWorks. http ://www.mathworks.com/products/3d-animation.

[6] H. Rayhane. *Surveillance des systèmes de production automatisés :Détection et Diagnostique*. PhD thesis, Institut national polytechnique de Grenoble, 2004.

[7] The MathWorks Inc Stateflow SF.Coder7. http ://www.mathworks.com/products/stateflow.

[8] The MathWorks Inc Simulink. http ://www.mathworks.com/products/simulink.

[9] The Math Work. *Stateflow For State Diagram Modeling*. User's Guide, 2001.

[10] The MathWorks Inc Real-Time Workshop. http ://www.mathworks.com/products/rtw.

[11] L. Yacoubi. *L'utilisation du DS1104 de DSPACE*. Ecole de technologie superieur université du QUÉBEC.

Annexes

FIGURE 5.5 – *La chaîne en 3D*

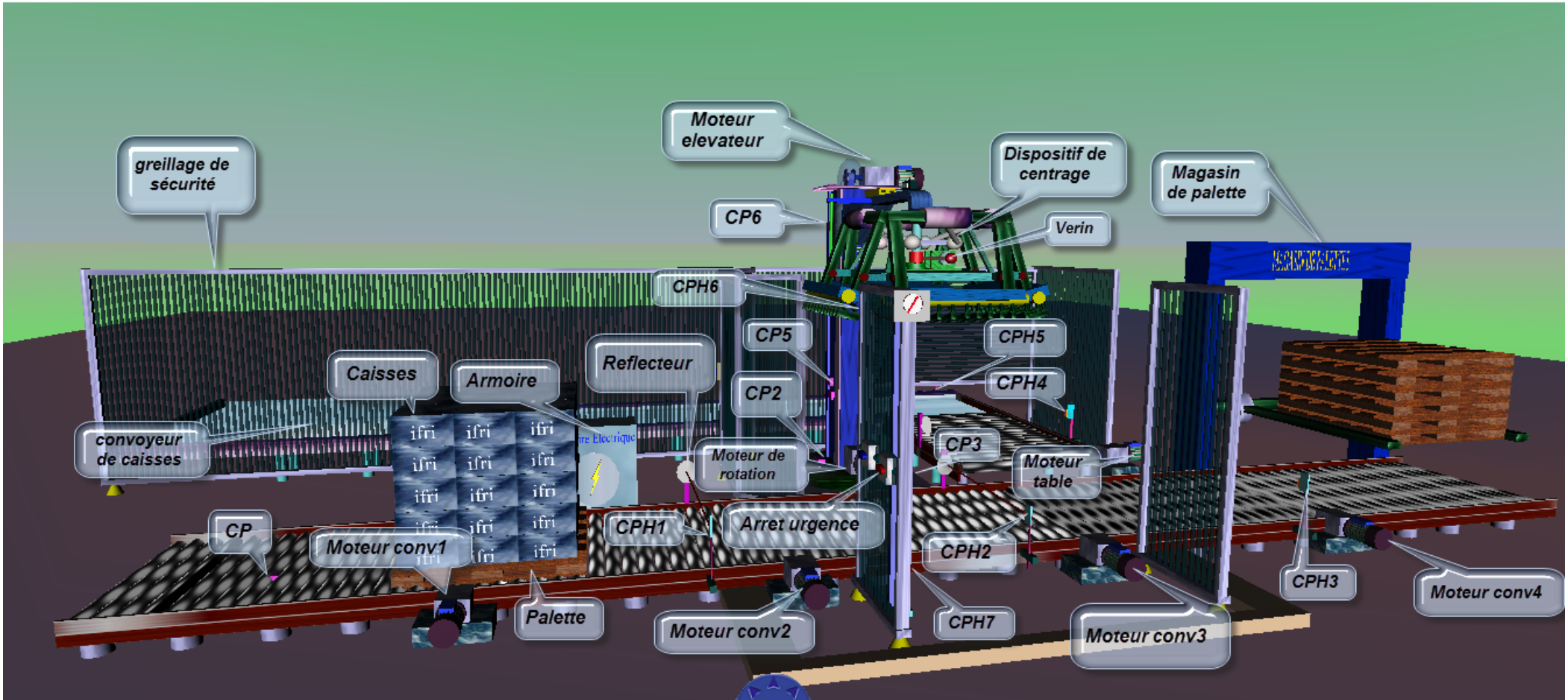

Moteur elevateur
Dispositif de centrage
Magasin de palette
greillage de sécurité
CP6
Verin
CPH6
CP5
CPH5
CP2
CPH4
Caisses
Armoire
Reflecteur
convoyeur de caisses
CP3
Moteur de rotation
Moteur table
CP
Moteur conv1
CPH1
Arret urgence
CPH2
CPH3
Moteur conv4
Palette
Moteur conv2
CPH7
Moteur conv3

FIGURE 5.6 – *Le schéma bloc Simulink avec VR.Sink*

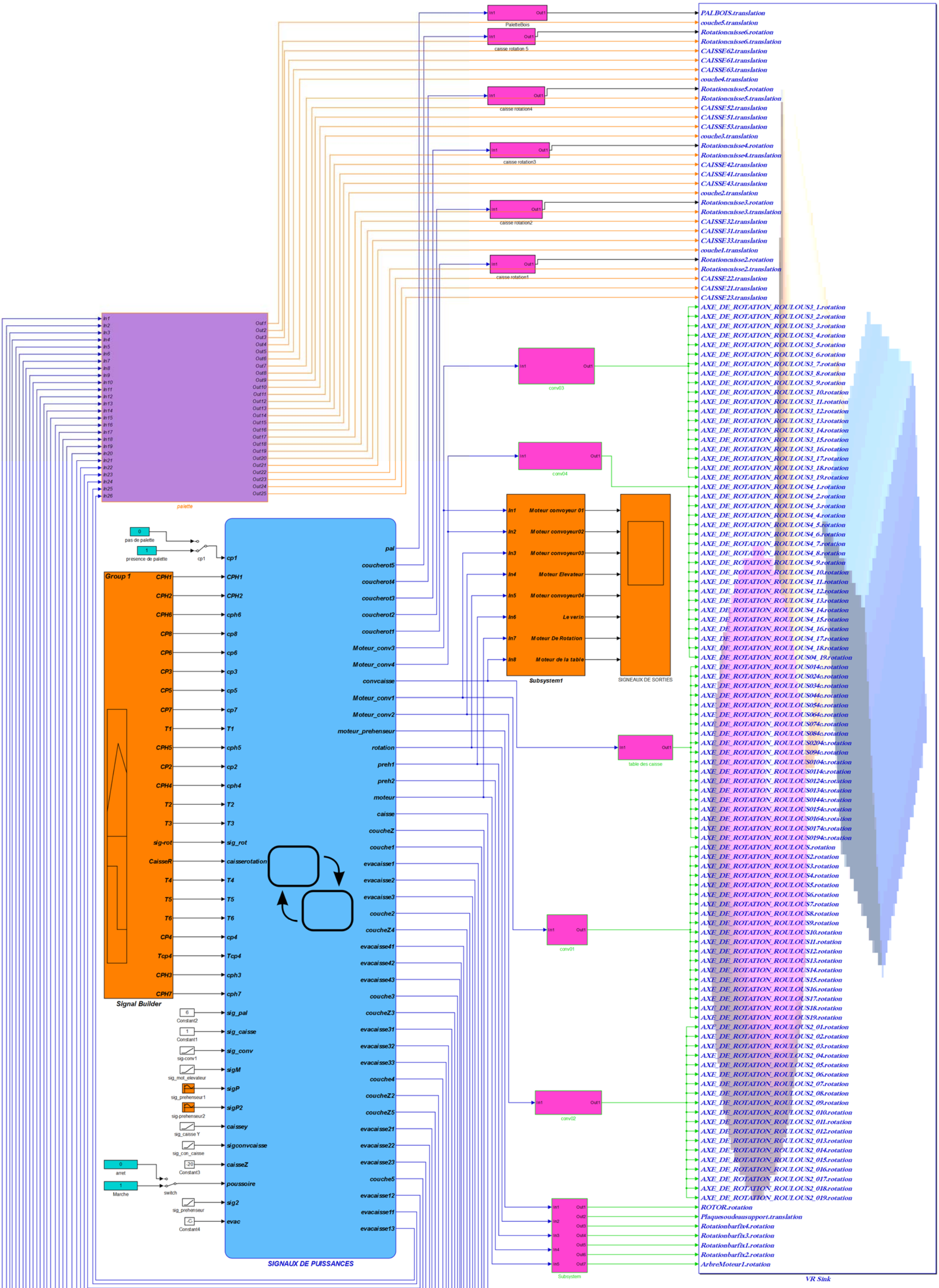
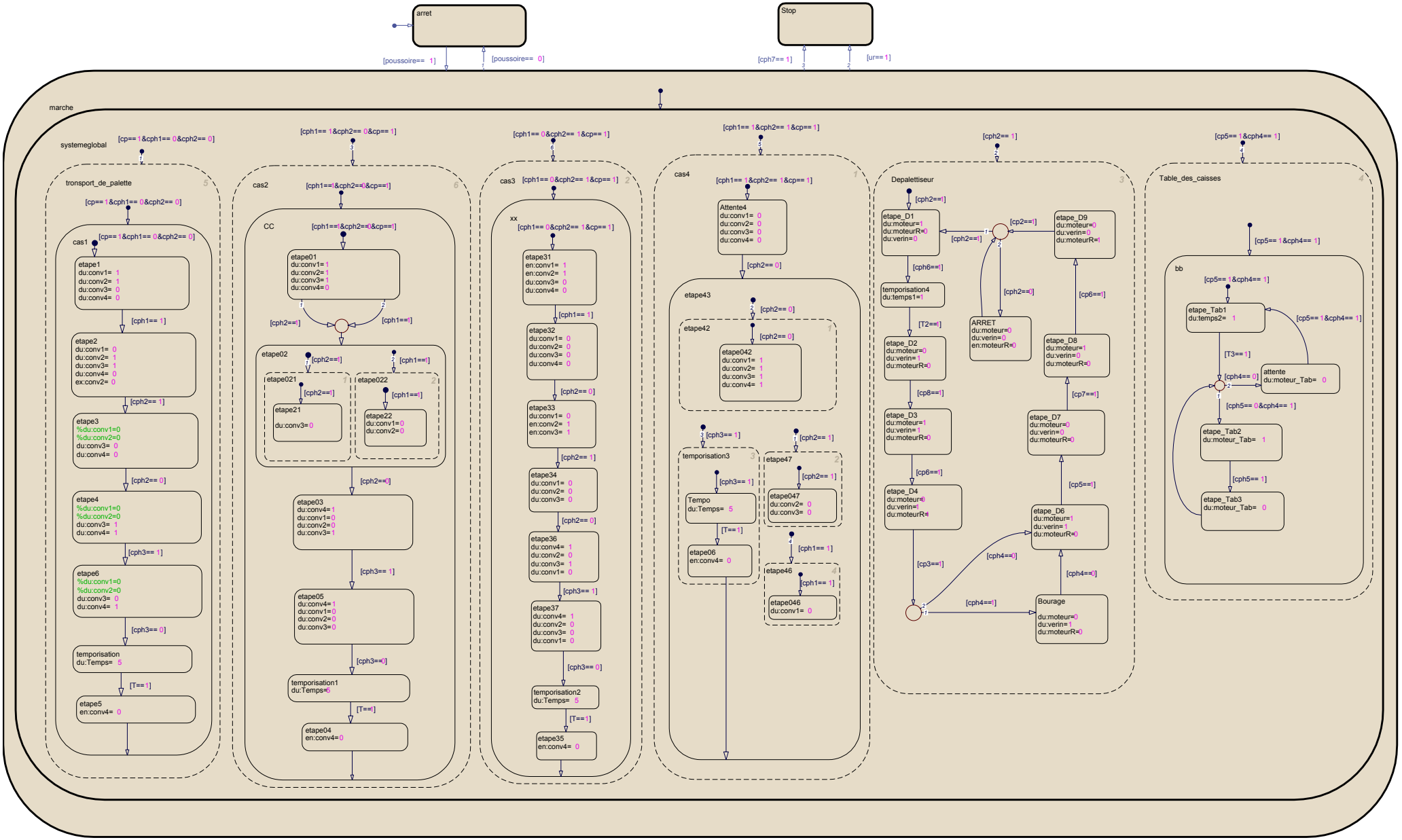

PaletteBois
caisse rotation 5
caisse rotation4
caisse rotation3
caisse rotation2
caisse rotation1
palette
pas de palette
presence de palette
cp1
Group 1
Signal Builder
Constant2
Constant1
sig-conv1
sig_mot_elevateur
sig_prehenseur1
sig-prehenseur2
sig_caisse Y
sig_con_caisse
arret
Marche
switch
Constant3
Constant4
sig_prehenseur
SIGNAUX DE PUISSANCES
conv03
conv04
conv01
conv02
table des caisse
Moteur convoyeur 01
Moteur convoyeur02
Moteur convoyeur03
Moteur Elevateur
Moteur convoyeur04
Le verin
Moteur De Rotation
Moteur de la table
Subsystem1
SIGNEAUX DE SORTIES
Subsystem
VR Sink

FIGURE 5.7 – *Le diagramme Stateflow du programme global*


arret
[poussoire== 1]
[poussoire== 0]
Stop
[cph7== 1]
[ur== 1]
marche
systemeglobal
[cp== 1&cph1== 0&cph2== 0]
tronsport_de_palette
cas1
etape1
etape2
etape3
etape4
etape6
temporisation
etape5
cas2
[cph1== 1&cph2== 0&cp== 1]
CC
etape01
etape02
etape021
etape21
etape022
etape22
etape03
etape05
temporisation1
etape04
cas3
xx
etape31
etape32
etape33
etape34
etape36
etape37
temporisation2
etape35
cas4
Attente4
etape43
etape42
etape042
temporisation3
Tempo
etape06
etape47
etape047
etape46
etape046
Depalettiseur
etape_D1
etape_D2
etape_D3
etape_D4
ARRET
etape_D9
etape_D8
etape_D7
etape_D6
Bourage
Table_des_caisses
bb
etape_Tab1
attente
etape_Tab2
etape_Tab3

Résumé

L'étude présentée dans ce mémoire s'inscrit dans le cadre d'application des techniques de l'automatisme industriel, pour l'automatisation et simulation en 3D d'une partie d'une chaîne de production dans le secteur de manutention dans le cas de la SARL Ifri.

Notre démarche consiste à définir les objets constituant le système étudié d'une part, et décrire l'ensemble des tâches de différents organes . Ensuite, à partir de l'outil Simulink 3D(V-Realm Builder) nous avons crée un modèle Simulink pour animer la scène virtuelle(la partie opérative), pour établir le modèle de commande de cette dernière, nous avons fait appel à un outil structurel adapté à la description et la commande des systèmes à événement discret qui est le 'Stateflow' .

En premier abord, on a présenté l'outil 3D dans l'environnement MATLAB avec ***Simulink 3D*** *(*toolbox*). Ensuite, on a décrit l'outil de l'automatisation et de supervision* ***Stateflow****, Après, dans un premier temps ,on a décrit le système réel à automatiser, puis on a conçu en* ***3D*** *des différents organes constituant le système. Ensuite, on a programmé et simulé la partie de la chaîne de production en temps réel avec l'outils Simulink/Stateflow. Enfin, on a passé à l'implémentation du modèle Simulink/Stateflow à partir de la carte dSPACE et sa validation en temps réel.*

Mots clés : Stateflow, V-realm Builder, VRML Toolbox, dépalettiseur, Transport de palettes, palettisation, Simulink 3D, dSPACE, ds1104.