Apport d'Altarica pour la sécurité du binage autonome

Jean-Loup Farges ONERA THE FRENCH AEROSPACE LAB

Pascal Schmidt naïo Technologies

Plan

- Introduction
- Le robot
  - Opération
  - Sécurité
- Le langage Altarica
- Les choix de modélisation
- Le modèle Altarica
  - Modèle externe
  - Modèle d’architecture
- La génération d’arbres
  - Sortie du champ
  - Collision
- Conclusion

Introduction

- Présence accrue de robots mobiles dans notre environnement → adoption de méthodes de sécurité par les fabricants
- Projet 

  - Financement UE Horizon 2020 numéro 644400
  - Expériences de dissémination de méthodes de sécurité
    - Analyse de sécurité basée modèle pour le robot Oz de Naio Technologies

Le robot

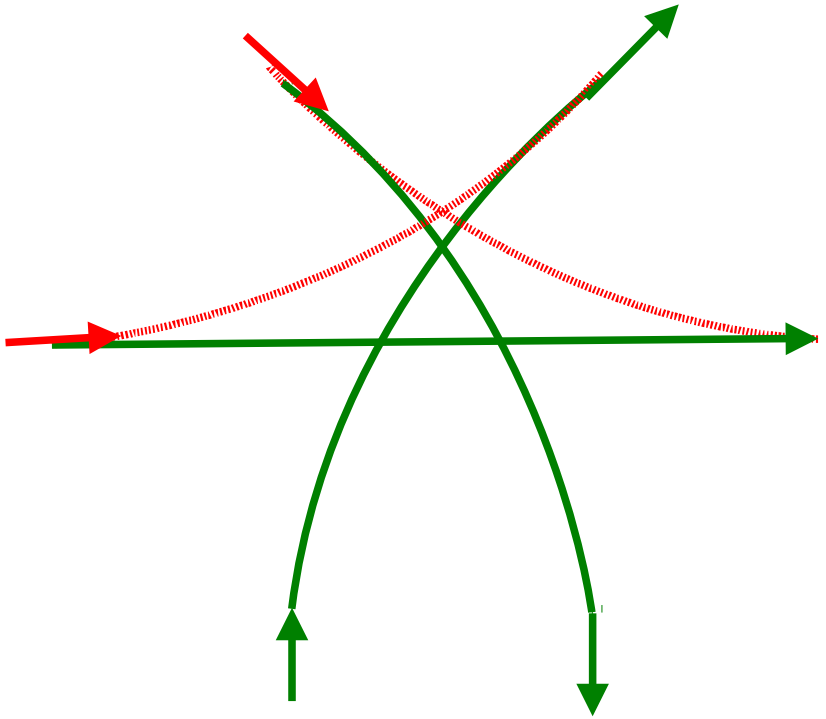
- Opération
  - Désherber les rangées de culture

Le robot

- Opération
  - Se guide dans les rangées à partir d’un lidar et de caméras

Le robot

- Opération
  - Une rangée, demi-tour, la rangée suivante
  - Demi-tour en cinq étapes



Le robot

- Opération

Le robot

- Sécurité aujourd'hui
  - Peu dangereux
    - Vitesse maximale : 1,3 km/h
    - Masse : 130 kg
  - Conforme aux directives CE
    - "machine"
    - "limitation de certaines substances dangereuses dans les équipements électriques et électroniques"
    - "compatibilité électromagnétique (CEM)"
  - Cas de panne
    - Collision
    - Sortie du champ à désherber
  - Sécurités opérationnelles mises en place
    - Bouton arrêt d'urgence
    - Pare-chocs sensibles
    - Grillage ou fossé autour des champs
    - Surveillance humaine...
- Sécurité dans le futur ?
  - Analyser la robustesse du robot à l'aide d'arbres de défaillance

Le langage Altarica

- Modélisation de systèmes à événements discrets
- Composants (*node*)
  - Flux (*flow*) d'entrée (*in*) et sortie (*out*)
  - Variables d'état (*state*)
    - Valeur initiale (*init*)
  - Événements (*event*)
  - Transitions gardées (*trans*)
  - Calcul de sortie (*assert*)

```
node NaioOz_OtherHardware_SignalTransmitter
 flow
  input_signal : bool : in ;
  output_signal : bool : out ;
 state
  out_of_service : bool ;
 event
  fail ;
 init
   out_of_service := false ;
```

```
trans
out_of_service = false |- fail -> out_of_service := true;
assert
output_signal = (input_signal and (not out_of_service));
```

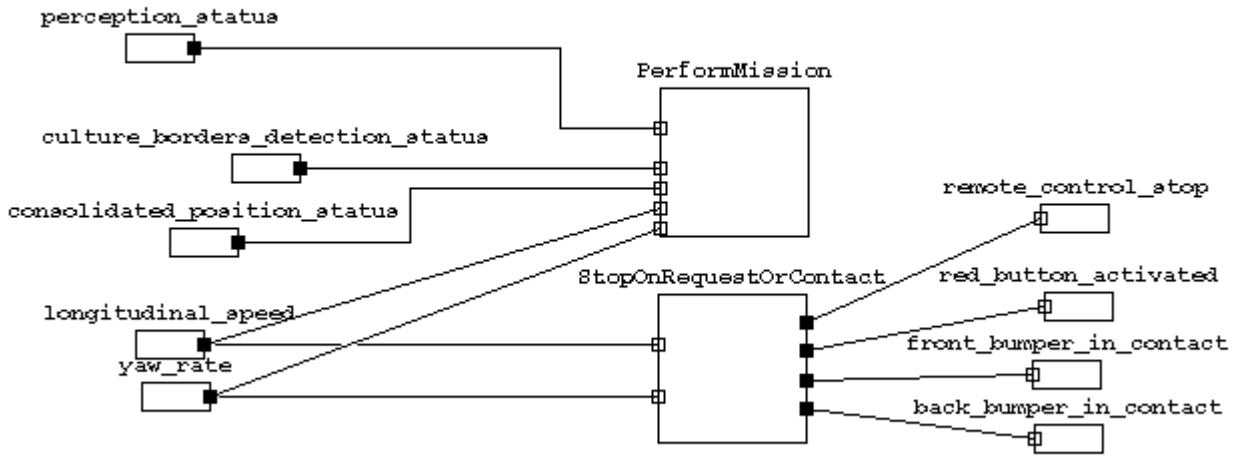
- Hiérarchie de composants dans des équipements
- Les flux sont des formules d'états
- Synchronisation d'événements

Les choix de modélisation

- Ne pas modéliser ce qui est jugé sans lien avec les cas de panne
  - Maniement de l’outil
- Séparer les vues
  - Un modèle externe pour exprimer :
    - L’interaction du robot avec son environnement
    - Les exigences et cas de panne
  - Un modèle interne pour exprimer :
    - L’architecture fonctionnelle
    - L’architecture matérielle
- Lier les modèles interne et externe
  - Variables
  - Événements synchronisés

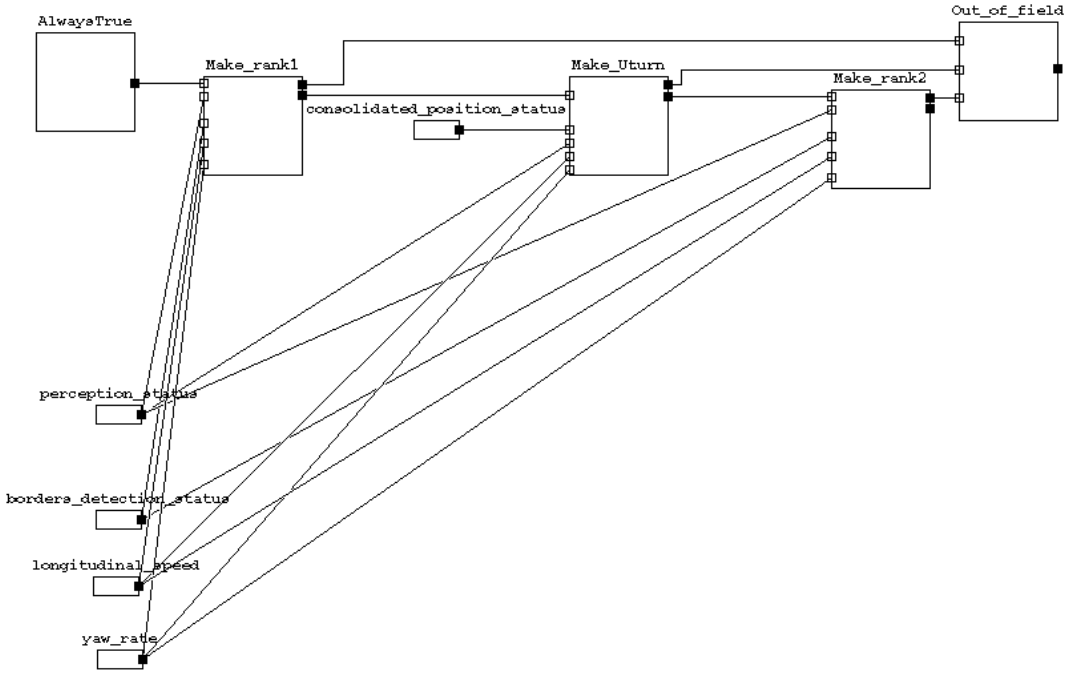
Le modèle Altarica – externe

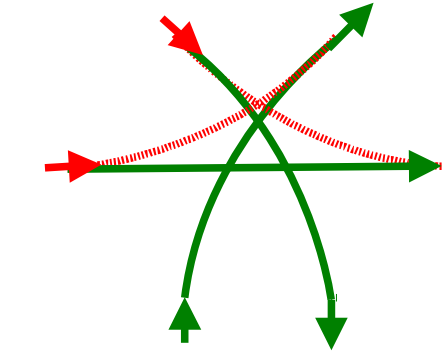
- Interaction de Oz avec son environnement
- Deux fonctions principales
  - Réaliser la mission
  - S’arrêter sur demande ou contact
- Types de variables
  - Booléennes : Pour les statuts vrai = pas de panne – faux = panne
  - Vitesses : Quantificateurs linguistiques {NL,NS,ZE,PS,PL}



Le modèle externe – Réaliser la mission

- Enchaînement de phases
  - Opportunité de sortir du champ dans chaque phase
  - Contiennent des sous phases
- Les sous phases contiennent des activités de déplacement avec :
  - des conditions sur les vitesses pour les démarrer et les faire
  - des conditions sur les statuts pour les terminer





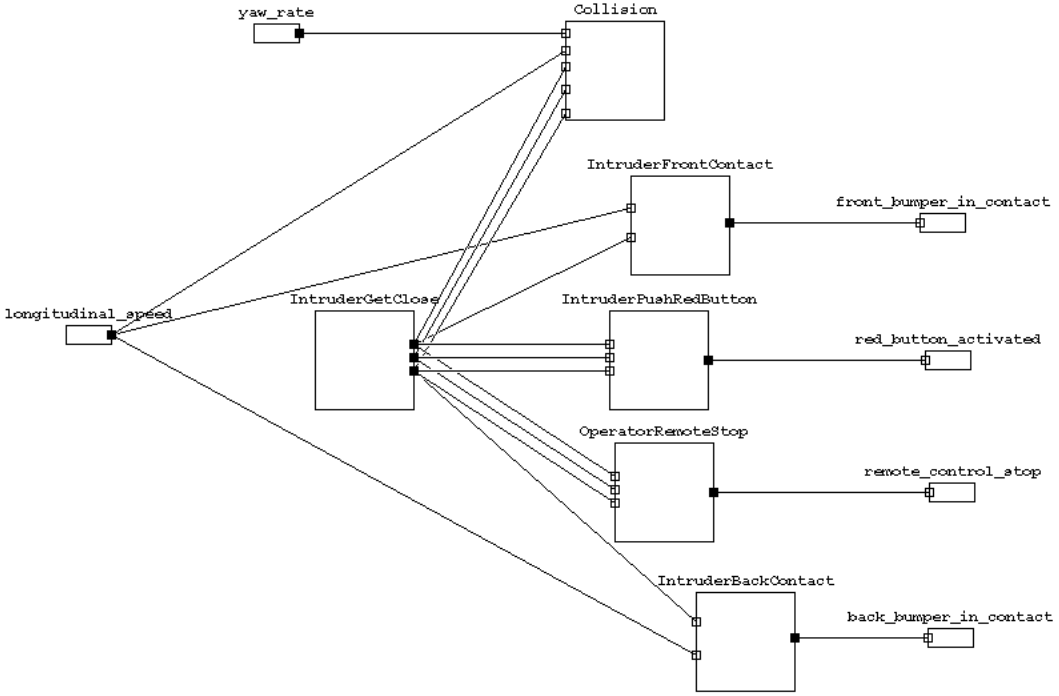
Rangée précédente

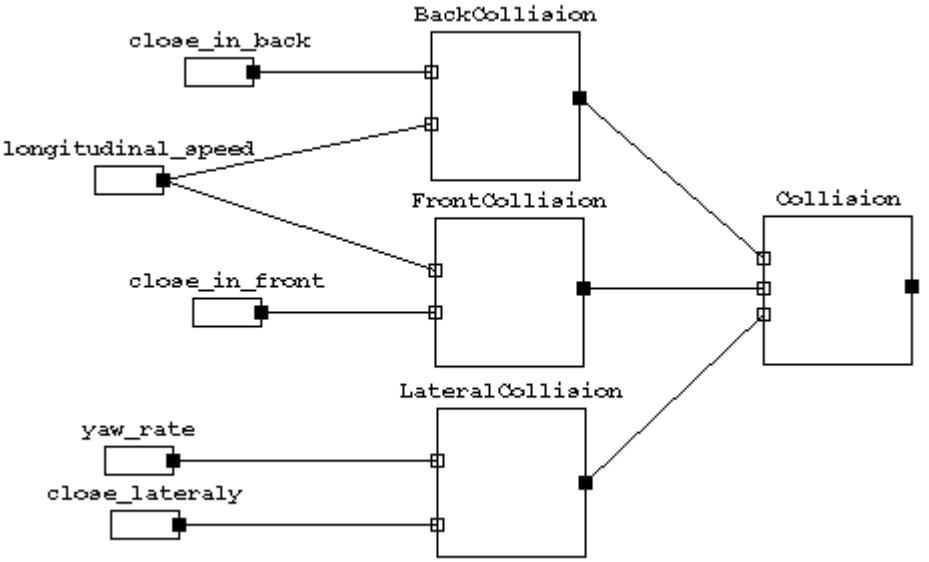
Rangée suivante

Demi tour

Le modèle externe – S’arrêter sur demande ou contact

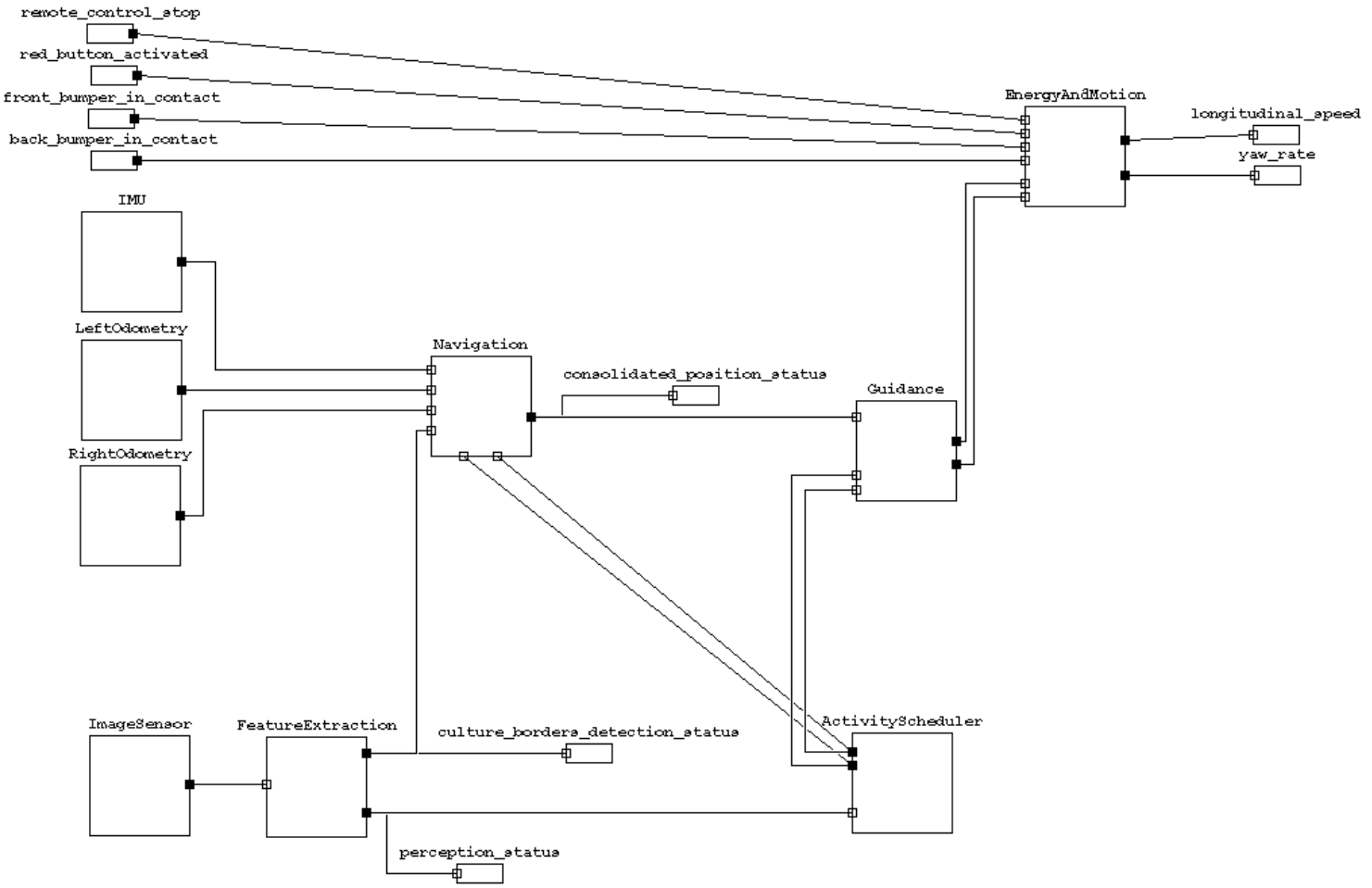
- Faire la différence entre
  - Collision : exemple vitesse longitudinale > 0 après un contact “pare-choc avant”
  - Simple contact : exemple contact “pare-choc avant” provoqué par une vitesse longitudinale > 0
- Événement “mise à jour”
  - Permet la mise à jour de la vitesse après contact pare-choc
  - N’est ni un événement redouté, ni une condition adverse, ni un élément de scénario → Dirac(0)





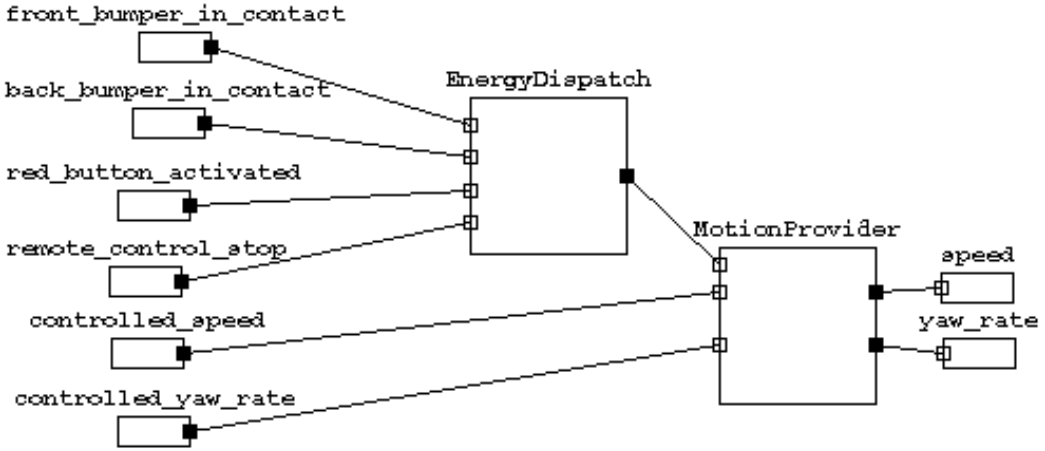
Le modèle Altarica – architecture

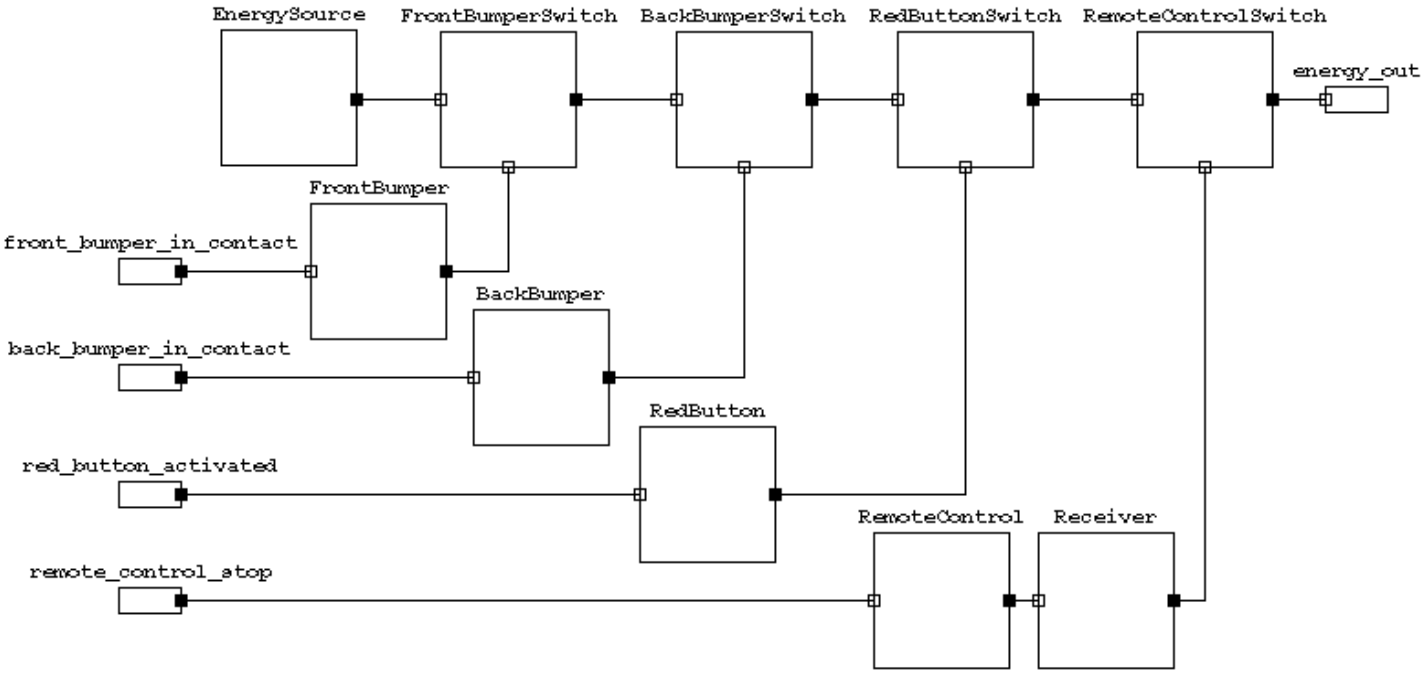
- Fonction de distribution d'énergie et de commande du déplacement
- Fonctions de perception, navigation, guidage et séquencement
- Synchronisations – fin de :
  - phase
  - sous-phase



Le modèle d'architecture – Énergie et commande du déplacement

- Partie de l'architecture en interaction avec l'exigence de non collision
- Ne comprend que des événements redoutés



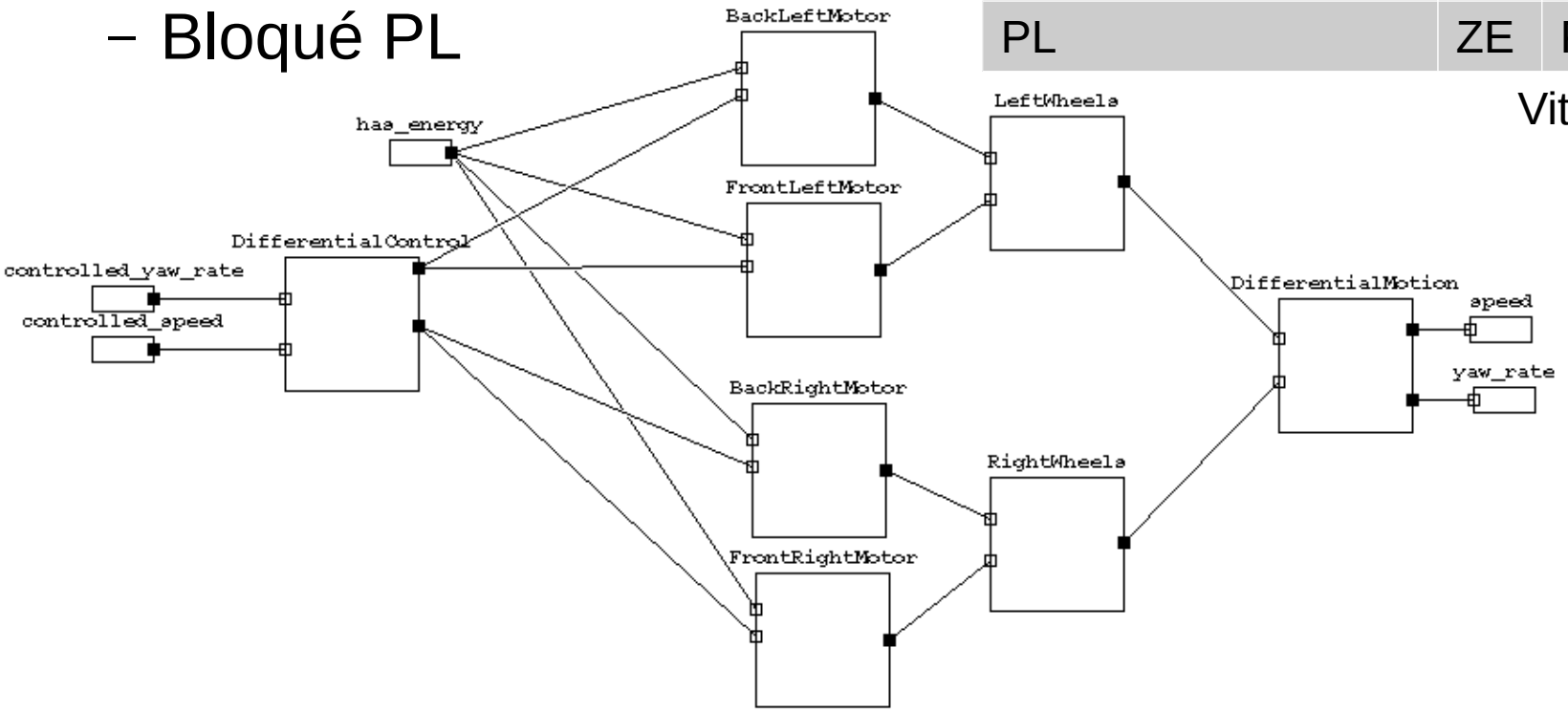


Le modèle d'architecture – Commande du déplacement

- Calcul qualitatif
- Trois pannes moteur
  - Bloqué ZE
  - Bloqué NL
  - Bloqué PL

First speed \ Second speed	NL	NS	ZE	PS	PL
NL	NL	NS	NS	NS	ZE
NS	NS	NS	NS	ZE	PS
ZE	NS	NS	ZE	PS	PS
PS	NS	ZE	PS	PS	PS
PL	ZE	PS	PS	PS	PL

Vitesse moyenne



Le modèle d'architecture – Autres fonctions

- Capteurs
  - IMU, Odométrie droite et gauche, Senseur d'image
  - État interne : hors service – événement : panne
- Navigation
  - Position consolidé disponible et correcte = (mode correct) et
    - Soit en demi-tour et statut IMU et statut Odométrie droite et statut Odométrie gauche
    - Soit en rangée et statut détection de culture
- Extraction de caractéristiques
  - Poteau
  - Rangées de plantations
- Guidage
  - Sous phases du demi-tour
  - Événement "fin de sous-phase"
  - Consignes en vitesses
- Ordonnancement
  - États :
    - En demi-tour ou en rangée
    - Phase erronée
  - Sortie vers le guidage en conséquence

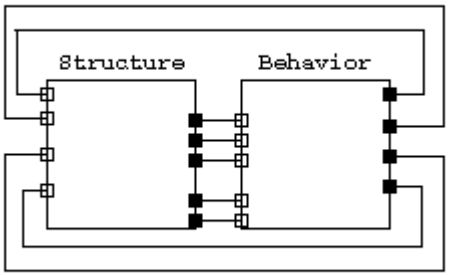
"fin de phase"

Statut détection poteau	En rangée avant événement	En rangée après événement	Phase erronée
Vrai	Vrai	Faux	idem
Vrai	Faux	Vrai	idem
Faux	Vrai	Vrai	Vrai
Faux	Faux	Faux	Vrai

Réaction de l'ordonnancement à l'événement "fin de phase"

La génération d’arbres - sortie du champ

- Connexion des modèles interne et externe
- Recherches de séquences
  - Répétitions possibles
  - A partir de l’état de début de mission
    - Jusqu’à 7 événements
  - A partir d’un état intermédiaire :
    - Au début de la marche avant du demi-tour
    - Simulation en nominal jusqu’au début de sous phase → état intermédiaire
    - Jusqu’à 6 événements



La génération d’arbres - sortie du champ

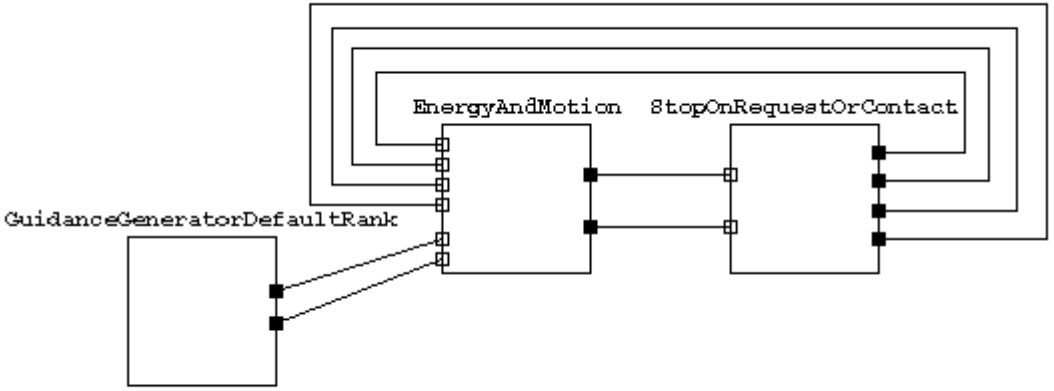
- Temps de calcul et coupes :
  - De l’état initial

Nombre d’événements	Temps de calcul	Coupes
2	0 s	2 ordre 2
3	0 s	2 ordre 2 + 3 ordre 3
4	8 s	2 ordre 2 + 3 ordre 3
5	1 m 48 s	2 ordre 2 + 3 ordre 3
6	42 m 33 s	2 ordre 2 + 3 ordre 3
7	16 h 15 m 64 s	2 ordre 2 + 3 ordre 3

  - De l’état intermédiaire pour 6 événements
    - 24 minutes 12 secondes – 3 coupes d’ordre 2 + 2 coupes d’ordre 4
- Coupes minimales d’ordre 1 après élimination des événements nominaux
  - Scénario : démarrage de rang, puis fin de sous phase ou démarrage de dernière sous phase de demi-tour, puis fin de phase
    - Panne senseur d’image ou Panne traitement d’image – probablement le plus critique
  - Scénario : démarrage de rang, puis fin de sous phase, puis démarrage de demi-tour, puis fin de sous phase ou démarrage de sous phase de demi-tour et fin de sous phase
    - Panne IMU ou Panne odométrie droite ou Panne odométrie gauche

La génération d’arbres - collision

- Connexion
  - Sous modèle d’arrêt sur demande ou contact
  - Sous modèle commande du déplacement
  - Générateur pour les commandes de vitesse :
    - de cap et longitudinale
    - par défaut consigne pour le scénario “en rang”
    - accepte des événements de scénario :
      - Demi-tour phase 0
      - Demi-tour phase 1
      - Demi-tour phase 2
      - Demi-tour phase 3
      - Demi-tour phase 4



- Génération de séquences
  - Avec répétition
  - Configurations de scénario
    - (frontale – latérale – dorsale) x (en rang ou phase 2 – phase 0 ou 4 – phase 1 ou 3)
  - De 6 à 9 événements

La génération d’arbres - collision

- Temps de calcul

Nombre d’événements Scénario	6	7	8	9
Frontale – phase 0 ou 4	59 s	10 m 27 s	1 h 38 m 17 s	12 h 30 m 10 s
Frontale – phase 1 ou 3	1 m 40 s	12 m 35 s	2 h 09 m 24 s	18 h 48 m 04 s
Frontale – rang ou phase 2	1 m 51 s	17 m 18 s	2 h 41 m 38 s	11 h 06 m 43 s
Latérale – phase 0 ou 4	1 m 17 s	13 m 20 s	1 h 47 m 51 s	7 h 21 m 20 s
Latérale – phase 1 ou 3	1 m 11 s	12 m 20 s	1 h 50 m 16 s	7 h 07 m 21 s
Latérale – rang ou phase 2	1 m 30 s	10 m 28 s	1 h 13 m 31 s	8 h 51 m 09 s
Dorsale – phase 0 ou 4	1 m 45 s	12 m 47 s	2 h 00 m 15 s	16 h 47 m 31 s
Dorsale – phase 1 ou 3	1 m 42 s	10 m 34 s	1 h 27 m 33 s	11 h 16 m 10 s
Dorsale – rang ou phase 2	1 m 43 s	12 m 49 s	2 h 03 m 22 s	18 h 16 m 15 s

La génération d'arbres - collision

Scénario – Phase / sous-phase Arrivée intrus	Rang ou phase 2 du demi-tour	Phase 0 ou 4 du demi-tour	Phase 1 ou 3 du demi-tour
Frontale	24 coupes ordre 3 $S_{FB} \times S_{RB} \times S_{RC}$	24 coupes ordre 3 $S_{FB} \times S_{RB} \times S_{RC}$	168 ordre 5 + 96 ordre 6 $S_{FB} \times S_{RB} \times S_{RC} \times S_{V}$
Latérale	96 coupes ordre 3 $S_{L} \times S_{RB} \times S_{RC}$	12 coupes ordre 2 $S_{RB} \times S_{RC}$	12 coupes ordre 2 $S_{RB} \times S_{RC}$
Dorsale	384 ordre 6 + 96 ordre 7 $S_{BB} \times S_{RB} \times S_{RC} \times S_{A}$	168 ordre 5 + 96 ordre 6 $S_{BB} \times S_{RB} \times S_{RC} \times S_{R}$	24 coupes ordre 3 $S_{BB} \times S_{RB} \times S_{RC}$

- S_{FB} : Chaîne pare-choc avant = $\lambda_{FB}+\lambda_{FBS}$ – pare choc ou relais pare-choc
- S_{BB} : Chaîne pare-choc arrière = $\lambda_{BB}+\lambda_{BBS}$ – pare choc ou relais pare-choc
- S_{RB} : Chaîne bouton d'arrêt d'urgence = $\lambda_{RB}+\lambda_{RBS}+\lambda_{ID}$ – bouton ou relais bouton ou distraction intrus
- S_{RC} : Chaîne télécommande = $\lambda_{RC}+\lambda_{R}+\lambda_{RCS}+\lambda_{OD}$ – télécommande ou communication ou relais télécommande ou distraction opérateur
- S_{L} : Lacet intempestif du robot = 4 ($\lambda_{N}+\lambda_{Z}$) - un des quatre moteurs en négatif ou à zéro
- S_{A} : Marche arrière intempestive en tout droit = 4 λ_{N}^{3} + 12 $\lambda_{N}^{2}.\lambda_{Z}$ + 4 $\lambda_{N}.\lambda_{Z}^{3}$ - trois des quatre moteurs en négatif ou deux en négatif et un à zéro ou un en négatif et trois à zéro
- S_{V} : Marche avant intempestive du robot = 5 λ_{P}^{2} + 2 $\lambda_{P}.\lambda_{Z}$ + 2 $\lambda_{P}.\lambda_{Z}(\lambda_{P}+\lambda_{Z})$- un des moteurs de droite positif et l'autre à zéro ou les deux de droite positif ou un à droite positif et un à gauche positif ou les 2 de gauche positifs et un de droite à zéro ou 1 de gauche positif et les deux de droite à zéro
- S_{R} : Marche arrière intempestive en virage = 5 λ_{N}^{2} + 2 $\lambda_{N}.\lambda_{Z}$ + 2 $\lambda_{N}.\lambda_{Z}(\lambda_{N}+\lambda_{Z})$- un des moteurs de gauche négatif et l'autre à zéro ou les deux de gauche négatifs ou un à droite négatif et un à gauche négatif ou les 2 de droite négatifs et un de gauche à zéro ou 1 de droite négatif et les deux de gauche à zéro

Conclusion – Réalisation

- Modèles externes et internes
- Simulation et génération d'arbres pour les cas de panne :
  - sortie du champ
  - collision
- Remerciements à Dassault Aviation pour une licence temporaire Cecilia OCAS

Conclusion – Pistes d’amélioration

- Réalisme du modèle par rapport à Oz
  - Modéliser la carte et le processeur qui supportent le logiciel
  - Modéliser plus précisément la chaîne pare-chocs
  - Évaluer des valeurs pour les taux de panne
  - Considérer une mitigation de la sortie de champ par la télécommande
- Généralité de la méthode et facilité d’usage
  - Réduction du nombre d’événements nominaux
    - Mettre les phases de mission en parallèle
    - Utiliser les états initiaux pour les scénarios
  - Rendre génériques les types d’activités et de composants
    - Activité de déplacement
    - Composant capteur
  - S’abstraire des valeurs fonctionnelles
    - Tout qualifier en termes de statuts
  - Traiter automatiquement les coupes
    - Classer
    - Factoriser
    - Générer des formules logiques simples
- S’attaquer à Dino qui est plus dangereux