



# De la modélisation au prototypage rapide



Environnement LabVIEW

Matériel compactRIO

Jean-Philippe BRAUD Société Phalanx



**Phalanx**

La compétence à votre côté

# Sommaire

- **Présentation des outils de simulation sous LabVIEW**
  - Simulation module
  - Démonstration sous LabVIEW
- **Validation d'un algorithme en prototypage rapide**
  - Introduction au système de test : Compact RIO
  - Etude de cas réel
  - Apport du prototypage rapide d'algorithme
- **Questions/Réponses**

# L'intervenant

Fondateur de Phalanx le 1<sup>er</sup> avril 2011.

Développeur certifié depuis 2007.

Architecte certifié LabVIEW depuis 2013.

Développe et supervise la réalisation de bancs de test, Acquisition de données

Réalisation d'IHM, de contrôle-commande embarqués

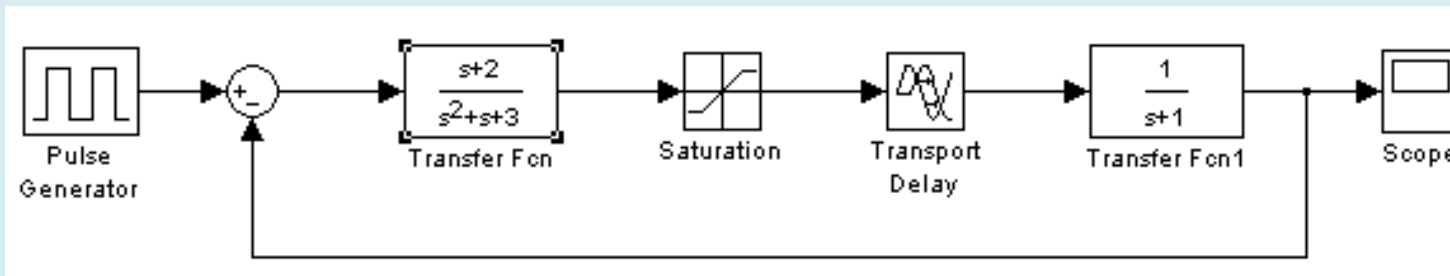
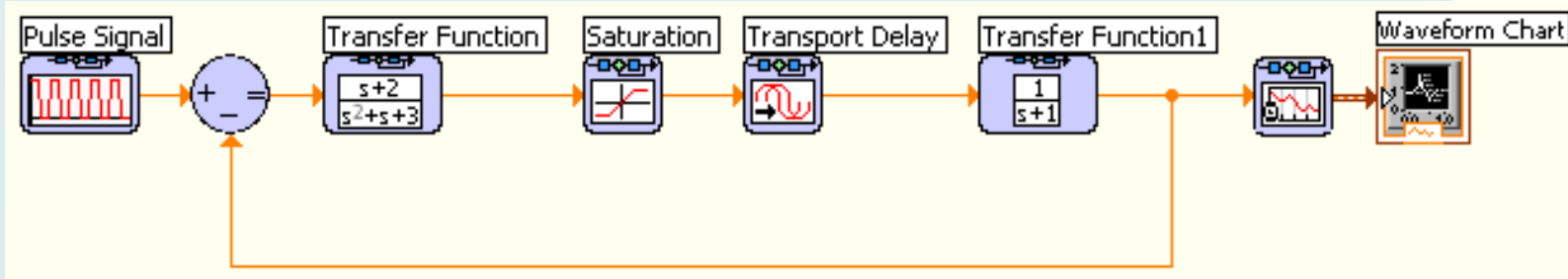
Utilisation majoritaire de LabVIEW

Intégration de code Matlab sous LabVIEW



# Simulation module

## Un environnement connu



# Simulation module

## ■ Ses fonctionnalités

Systèmes linéaires continus et discrets

Système non-linéaires

Par pas fixes, variables

Linéarisation

Visualisation 3D



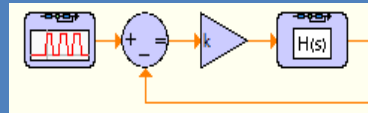


# Deploy to Hardware Through LabVIEW

MathScript RT  
Module

```
1 A = [0 1; -2 -4];  
2 eigs = eig(A);  
3 detA = det(A);
```

Control Design &  
Simulation Module



Your .m code

MATLAB®

Your .mdl code

Simulink®

Simulink Coder™

LabVIEW Real-Time

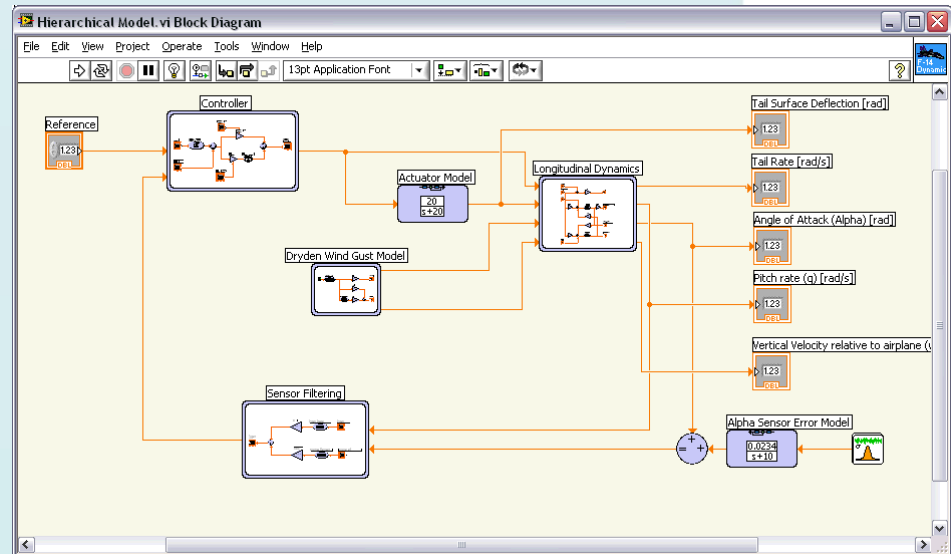
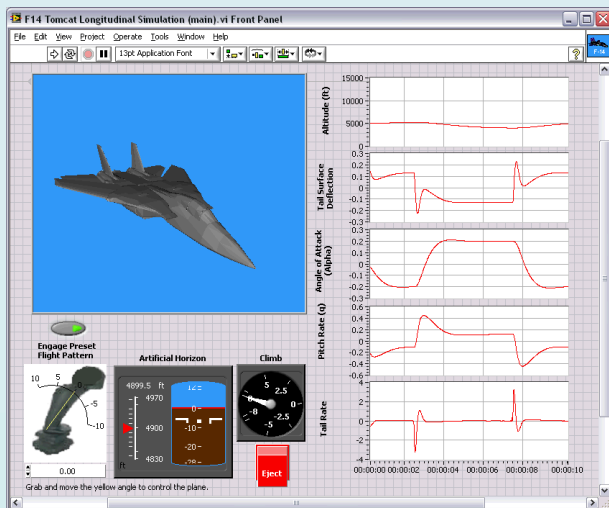
NI VeriStand

# Simulation module

## Utilisation

Simulation de système dynamique

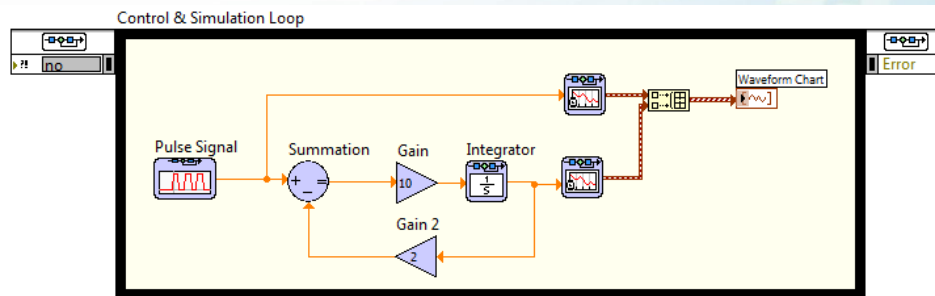
Implémentation temps réel pour du prototypage rapide ou Hardware In the Loop.  
(HIL)





# Simulation module

## Démonstration sous LabVIEW







# Validation d'un algorithme en prototypage rapide

Matériel RIO

Exemple d'utilisation

# Matériel RIO

- Un cœur en RT et un accès FPGA



```
void  
CreateRectPgn(80,160,80+  
jg.PtInRegion(point))  
  
double xOc,yOc;  
for (int i=0;i<NUNCITY;i++)  
xOc=80+RectSizeX*(xOc+  
yOc=160+RectSizeY*(yOc+  
pbn_rgnEllipse;  
- CreateEllipse
```



# Exemple d'utilisation

## Validation d'un algorithme en prototypage rapide

Besoin : Rétrofit et modernisation d'un banc Aéro.

Problème : Algorithme existant mal compris et demande client exigeante.

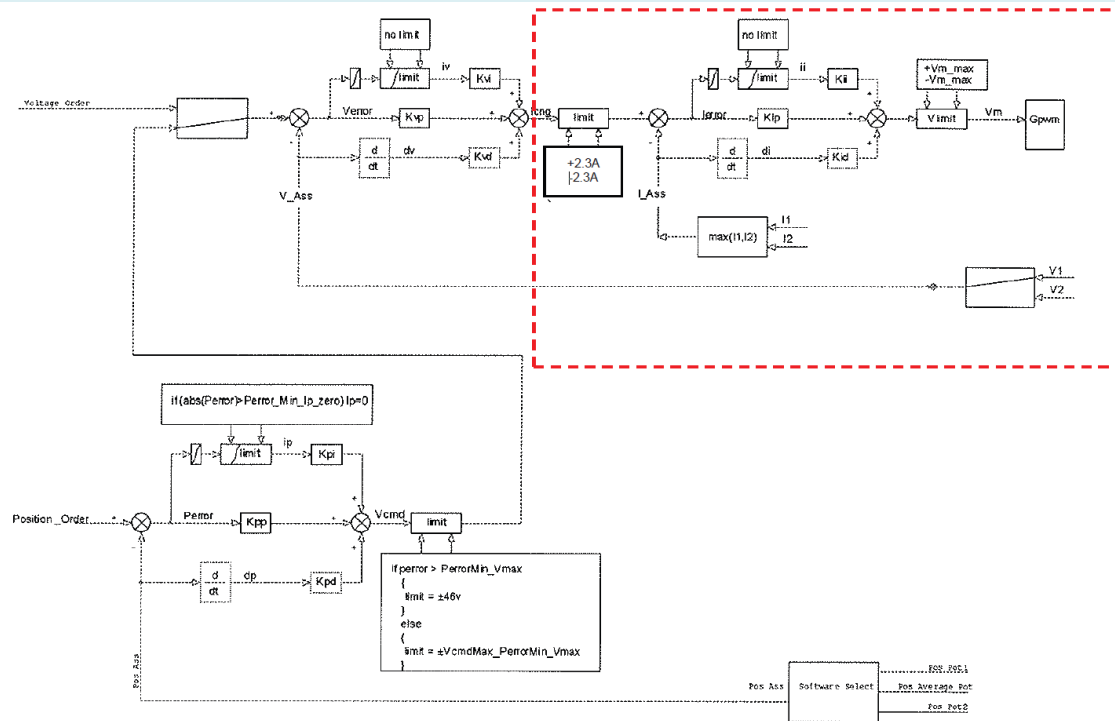
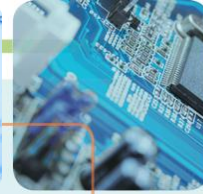
- 1 – Analyse du besoin
- 2 – Utilisation sous Real Time de l'algorithme de contrôle.
- 3 – Implémentation sous RT et FPGA.

# 1 – Analyse du besoin

```

1  CreateRectPgn(80,160,80+
2  jn.PtInRegion(point))
3
4  double x0c,y0c;
5  for (int i=0;i<NUMCITY;i++)
6  {
7      x0c=80+RectSizeX*(x0c+
8      y0c=160+RectSizeY*(y0c+
9      ->PgnEllipse;
10     ->CreateEllipse;

```



## Existant :

- 3 boucles PID en cascade
- position, courant, tension
- Régulation à 100Hz (10ms)

## Nouveau banc :

- Kis, Kps et Kds inconnus
- Trois vitesses distinctes
- 100Hz, 1kHz, 10kHz

## 2 – Algorithme en Temps Réel

Passage de l'algorithme « brut » sous LabVIEW avec le module Simulation.  
Tester l'idée et estimer les gains.

100Hz (10ms)

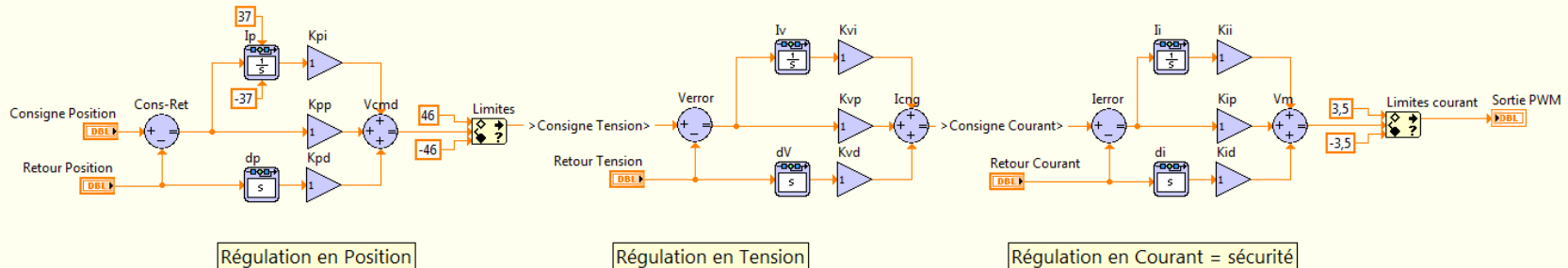
```
Signal  
CreateRectPgn(80,160,80+  
jgn.PtInRegion(point))  
  
double x0c,y0c;  
for (int i=0;i<NUNCITY;i++)  
  
x0c=80+RectSizeX*(xw+  
y0c=160+RectSizeY*(y+  
cRgn.rgnEllipse;  
- CreateEllip
```



Control & Simulation Loop

### Boucle de régulation en Position, Tension et Courant

[Covers: REQID\_E13069-SYS\_SES\_SW\_0310]  
[Covers: REQID\_E13069-SYS\_SES\_SW\_0330]  
[Covers: REQID\_E13069-SYS\_SES\_SW\_0350]





# 3 – Implémentation sous RT et FPGA

## Répartition des boucles :

- 100Hz sous RT pour limiter la charge sous FPGA,
- 1kHz sur le FPGA pour respecter les cadencements,
- 10KHz sur le FPGA.

## Prise sur le FPGA :

- 2 PWM,
- 2 PID,
- acquisitions de données,
- multiplexage,
- sécurités,
- Watchdog,

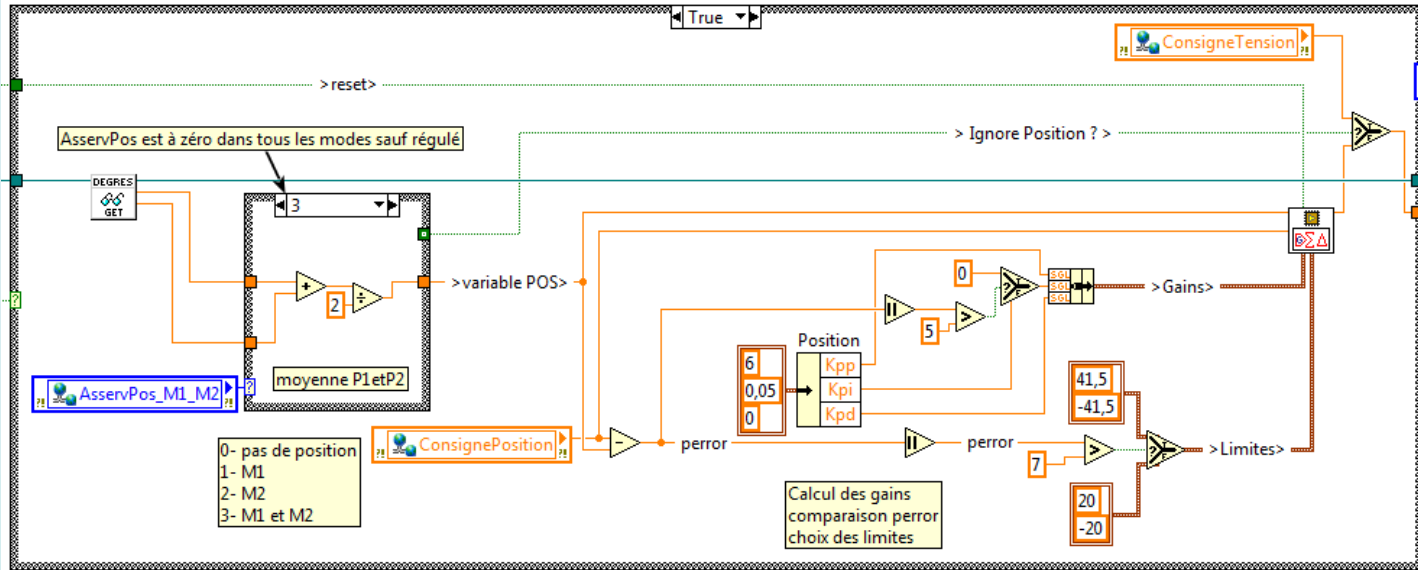
=>96% du FPGA utilisé !!!



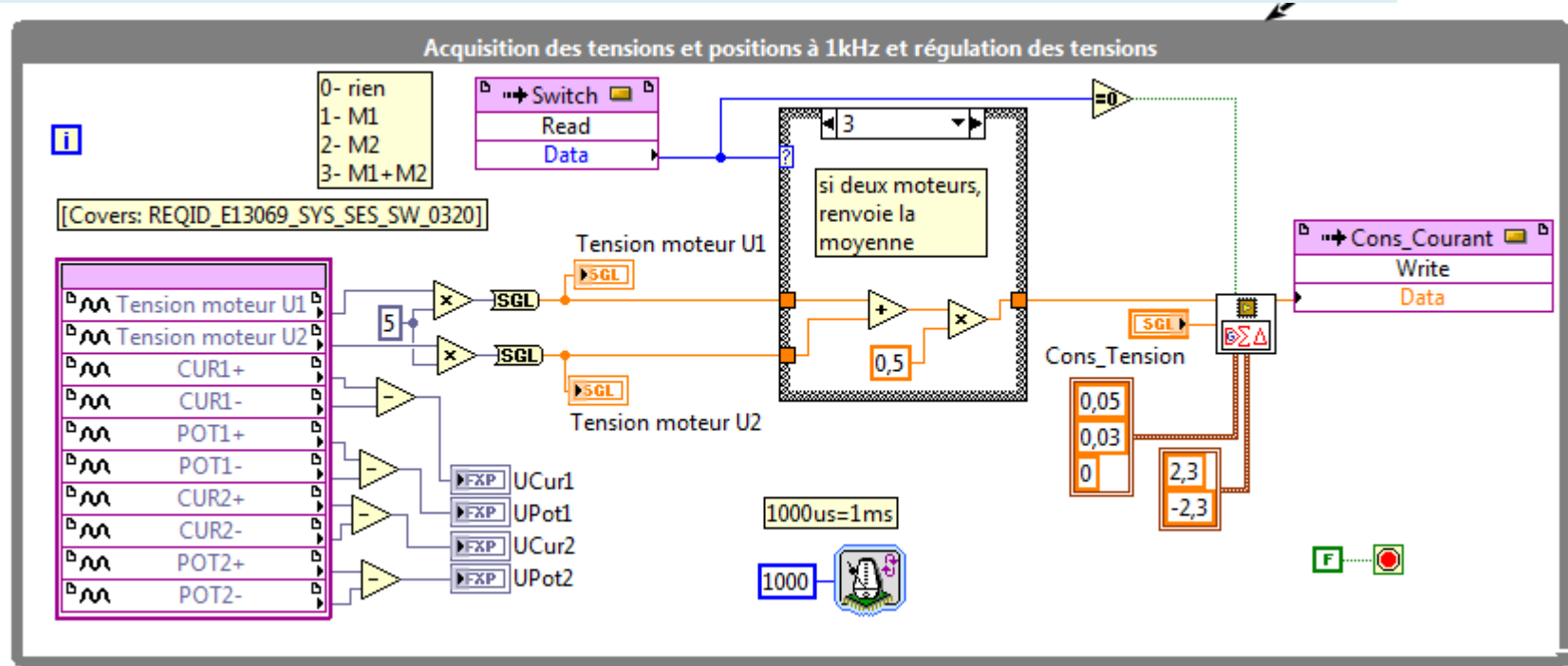
# 3 – Implémentation sous RT et FPGA

## Boucle asservissement en position 100Hz

[Covers: REQID\_E13069\_SYS\_SES\_SW\_0350]



# 3 – Implémentation sous RT et FPGA

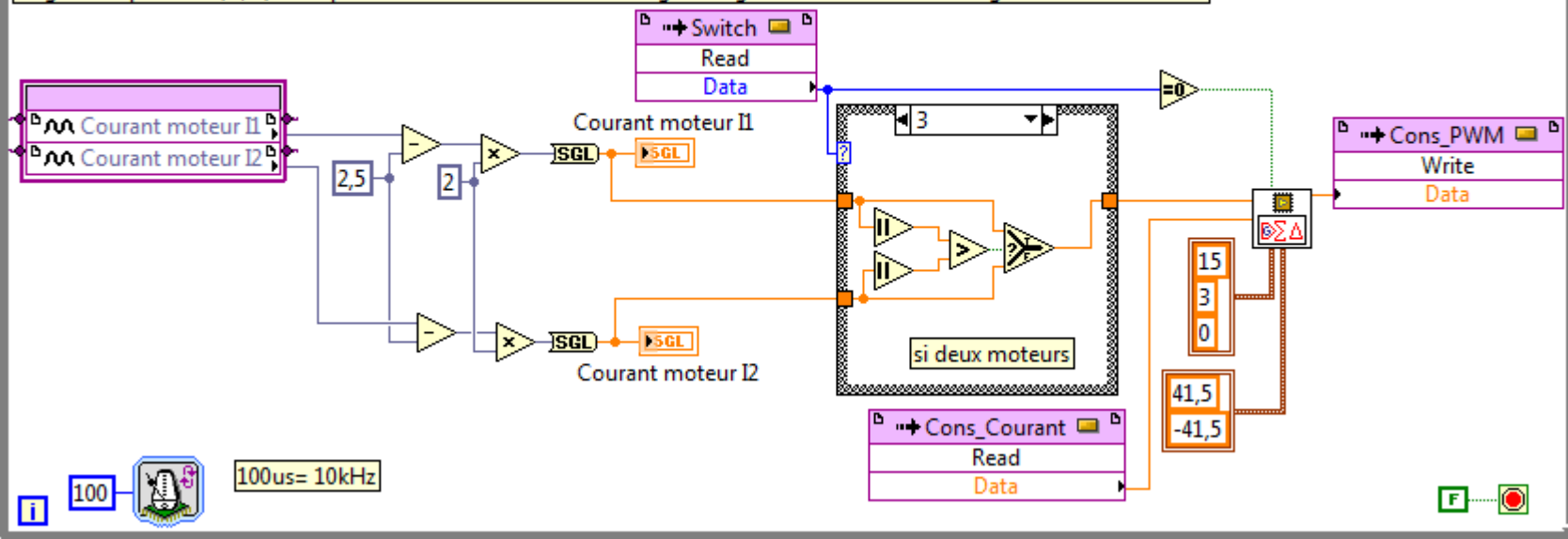


# 3 – Implémentation sous RT et FPGA

Acquisition et régulation des courants à 10kHz

[Covers: REQID\_E13069\_SYS\_SES\_SW\_0300]

l'algo n'est plus Max(I1,I2) mais prend en considération des consignes négatives et la sortie de la régulation 1kHz tension



```
Fig1:  
CreateRectFgn(80,160,80+  
gn.FtInRegion(point))  
  
void xOc,yOc;  
for (int i=0;i<NUNCITY;i++)  
xOc=80+RectSizeX*(xEm  
yOc=160+RectSizeX*(yEm  
-gn.FgnEllipse;  
- CreateEllipse
```



# Apport du prototypage rapide

## Gain de temps sur l'intégration du logiciel :

- Validation d'un algorithme, des limites d'un système, des IHMs et transfert des données.

## Gain de temps matériel:

- Validation de la mécanique
- Validation de l'électronique
- Le tout en système dégradé

On limite donc l'impact de l'intégration logicielle finale sur le projet global et les dépassements associés.





# Mot de la fin ?

Aujourd'hui on demande aux ingénieurs et techniciens de faire vite et bien au moindre coût... **Ces objectifs sont juste ridicules !**

Mais il faut les tenir au mieux.

Aussi, prototyper et réaliser des intégrations intermédiaires apparaît comme la bonne méthode.

 [www.captronic.fr](http://www.captronic.fr)

iD<sup>4</sup>for CAR

Moteur d'idées pour véhicules spécifiques et mobilité durable

 **NATIONAL  
INSTRUMENTS™**

 **SHERPA  
ENGINEERING**

 **FAAR®**  
2004 - 2014 *industry*

 **acsystème®**  
analyse & contrôle des systèmes



JESSICA FRANCE - 17, rue des Martyrs - 38054 GRENOBLE

Cedex 09

CAP'TRONIC, un programme financé par :

JESSICA FRANCE fondée par :



 **dgcis**  
direction générale de la compétitivité  
de l'industrie et des services



 **oséo**

