

La valeur ajoutée du calcul formel pour la modélisation de système physique

N. GACHADOIT, B. VIDALIE, Maplesoft France



**LMCS Printemps 2008
17 avril 2008**

Agenda

- Les principales techniques disponibles
- Exemples d'applications et apport du calcul formel
- Les nouvelles techniques :
 - Document mathématique interactif
 - Génération automatique d'équations
 - Production de code optimisé pour le HIL
- Démonstration:
 - Modélisation et pilotage d'un avion
 - Modélisation et pilotage d'un système électromécanique

Principales techniques disponibles

Formulation d'hypothèses

Expressions mathématiques

Simplification

Factorisation

Tri

Fonctions spéciales

Résolution analytique

Principales techniques disponibles

Calcul à précision fixée par l'utilisateur

Dérivée d'une fonction

Primitive d'une fonction

Unités physiques

Constantes physiques

Equations Différentielles

Système algébro différentiel

Principales techniques disponibles

Résolution de systèmes algébro-différentiels d'indice élevé

Résolution d'EDP

Dérivée d'un programme ou procédure

Calcul d'une cinématique inverse

Calcul d'une dynamique inverse

Moteur numérique directement sur les équations

Langage programmation mathématique

Principales techniques disponibles

Visualisation des équations et résultats au format Math 2D

Transformées

Représentation système :
termes d'amortissement en fonction des paramètres
physiques y compris pour les représentations discrètes

Modélisation physique : Utilisateurs Français de longue date chez

ASTRIUM Launch Vehicles

RENAULT

MBDA

CELLAR

DGA - CTSN

RVI -Volvo Trucks

CEA

ST Microelectronics

...

Exemples d'application et apport du calcul formel

Construction de modèles simplifiés **directement à partir de la physique**

Hypothèses sur la modélisation =
permettent de **réduire la taille**

Ecriture des hypothèses et formulations naturelles =
garder toute l'information sur le modèle

Simplification formelle = **sans compromis sur la précision**

Exemples d'application et apport du calcul formel

Formulation complète =
pas de sous entendu lié à un formalisme

Exemple : contraintes algébriques

Identification paramétrique =
identifier directement les paramètres physiques

Simplification des équations d'un système :
exemple modèle de moteur importé d'un schéma bloc

Simplification formelle : exemple sur un moteur

TorqueProbe1.flange_a.phi(t), TorqueProbe1.flange_b.phi(t), TorqueProbe1.flange_a.tau(t), TorqueProbe1.tau(t), TorqueProbe1.flange_b.tau(t), TorqueProbe1.tau(t), AngularVelocityProbe1.phi(t), AngularVelocityProbe1.flange_a.phi(t), D, AngularVelocityProbe1.flange_a.tau(t), DCmotor.Resistor1.r(t), DCmotor.Resistor1.p(t), DCmotor.Resistor1.n.r(t), D, DCmotor.Resistor1.p(t), DCmotor.Resistor1.n.t(t), DCmotor.Resistor1.t(t), DCmotor.Inertia1.flange_a.phi(t), DCmotor.Inertia1.phi(t), DCmotor.Inertia1.flange_b.phi(t), DCmotor.Inertia1.phi(t), DCmotor.Ground1.p.r(t), D, DCmotor.SignalVoltage1.r(t), DCmotor.SignalVoltage1.p.r(t), DCmotor.SignalVoltage1.n.r(t), D, DCmotor.SignalVoltage1.p.t(t), DCmotor.SignalVoltage1.n.t(t), DCmotor.SignalVoltage1.p.t(t), DCmotor.SignalVoltage1.n.t(t), DCmotor.Inductor1.p(t), DCmotor.Inductor1.p.r(t), DCmotor.Inductor1.n.p(t), D, DCmotor.Inductor1.p.t(t), DCmotor.Inductor1.n.t(t), DCmotor.Inductor1.t(t), DCmotor.Inductor1.p.t(t), DCmotor.EMF1.r(t), DCmotor.EMF1.p.r(t), DCmotor.EMF1.n.r(t), D, DCmotor.EMF1.p.t(t), DCmotor.EMF1.n.t(t), DCmotor.EMF1.t(t), DCmotor.Inertia1.flange_a.tau(t), DCmotor.flange_a.tau(t), D, DCmotor.Inertia1.flange_b.tau(t), DCmotor.flange_b.tau(t), DCmotor.SignalVoltage1.r(t), DCmotor.r(t), DCmotor.EMF1.flange_a.tau(t), DCmotor.Inertia1.flange_a.tau(t), D, DCmotor.EMF1.flange_b.tau(t), DCmotor.Inertia1.flange_a.tau(t), DCmotor.EMF1.flange_b.tau(t), DCmotor.Inductor1.n.t(t), D, DCmotor.EMF1.p.r(t), DCmotor.Inductor1.n.r(t), DCmotor.EMF1.n.t(t), DCmotor.SignalVoltage1.n.t(t), DCmotor.Ground1.p.t(t), D, DCmotor.EMF1.n.r(t), DCmotor.SignalVoltage1.n.r(t), DCmotor.SignalVoltage1.n.r(t), DCmotor.Ground1.p.r(t), DCmotor.SignalVoltage1.p.t(t), DCmotor.Resistor1.p.t(t), D, DCmotor.SignalVoltage1.p.r(t), DCmotor.Resistor1.p.r(t), IdealGear1.bearing.phi(t), D, D, IdealGear1.flange_a.tau(t), IdealGear1.flange_b.tau(t), IdealGear1.tau_support(t), IdealGear1.phi_a(t), IdealGear1.flange_a.phi(t), IdealGear1.bearing.phi(t), IdealGear1.phi_b(t), IdealGear1.flange_b.phi(t), IdealGear1.bearing.phi(t), Inertia2.phi(t), Inertia2.flange_b.phi(t), Inertia2.phi(t), SpringDamper1.phi_rel(t), SpringDamper1.flange_b.phi(t), SpringDamper1.flange_a.phi(t), SpringDamper1.flange_b.tau(t), SpringDamper1.tau(t), SpringDamper1.flange_a.tau(t), SpringDamper1.tau(t), TorqueProbe2.flange_a.phi(t), TorqueProbe2.flange_b.phi(t), TorqueProbe2.flange_a.tau(t), TorqueProbe2.tau(t), TorqueProbe2.flange_b.tau(t), AngularVelocityProbe2.phi(t), AngularVelocityProbe2.flange_a.phi(t), D, AngularVelocityProbe2.flange_a.tau(t), D, SpeedSensor1.flange_a.tau(t), Feedback1.r(t), Feedback1.u(t), Feedback1.z(t), Inertia2.flange_a.tau(t), SpeedSensor1.flange_a.tau(t), D, Inertia2.flange_b.phi(t), SpeedSensor1.flange_a.phi(t), Feedback1.z(t), SpeedSensor1.w(t), Z1.r(t), DCmotor.r(t), Feedback1.p(t), Z1.u(t), Stp1.r(t), Feedback1.u(t), AngularVelocityProbe2.flange_a.tau(t) + SpringDamper1.flange_b.tau(t) + TorqueProbe2.flange_a.tau(t) - D, AngularVelocityProbe2.flange_a.phi(t), SpringDamper1.flange_b.phi(t), SpringDamper1.flange_b.phi(t), TorqueProbe2.flange_a.phi(t), TorqueProbe2.flange_b.tau(t) + Inertia2.flange_a.tau(t) - D, TorqueProbe2.flange_b.phi(t), Inertia2.flange_a.phi(t), IdealGear1.flange_b.tau(t) + SpringDamper1.flange_a.tau(t) - D, IdealGear1.flange_b.phi(t), SpringDamper1.flange_a.phi(t), TorqueProbe1.flange_b.tau(t) + IdealGear1.flange_a.tau(t) - D, TorqueProbe1.flange_b.phi(t), IdealGear1.flange_a.phi(t), AngularVelocityProbe1.flange_a.tau(t) + DCmotor.flange_b.tau(t) + TorqueProbe1.flange_a.tau(t) - D, AngularVelocityProbe1.flange_a.phi(t), DCmotor.flange_b.phi(t), DCmotor.flange_b.phi(t), TorqueProbe1.flange_a.phi(t), IdealGear1.bearing.tau(t) - D, Stp1.r(t) $\begin{cases} D, t < D \\ 1, otherwise \end{cases}$ AngularVelocityProbe1.phi(t) - $\frac{d}{dt}$ AngularVelocityProbe1.phi(t), DCmotor.Inertia1.w(t) - $\frac{d}{dt}$ DCmotor.Inertia1.phi(t), DCmotor.Inertia1.t(t) - $\frac{d}{dt}$ DCmotor.Inertia1.w(t), DCmotor.EMF1.w(t) - $\frac{d}{dt}$ DCmotor.EMF1.flange_b.phi(t), Inertia2.w(t) - $\frac{d}{dt}$ Inertia2.phi(t), Inertia2.t(t) - $\frac{d}{dt}$ Inertia2.w(t), SpringDamper1.w_rel(t) - $\frac{d}{dt}$ SpringDamper1.phi_rel(t), AngularVelocityProbe2.w(t) - $\frac{d}{dt}$ AngularVelocityProbe2.phi(t), SpeedSensor1.w(t) - $\frac{d}{dt}$ SpeedSensor1.flange_a.phi(t), DCmotor.Resistor1.t(t) + DCmotor.Inductor1.p.t(t) - D, DCmotor.Resistor1.n.r(t) - DCmotor.Inductor1.p.r(t), SpringDamper1.phi_rel(t) - D, DCmotor.Inertia1.w(D) - D, Z1.a(D) - D, Inertia2.w(D) - D, AngularVelocityProbe2.phi(D) - D, DCmotor.Inertia1.phi(D) - D, Inertia2.phi(D) - D, DCmotor.Inductor1.t(D) - D, SpeedSensor1.flange_a.phi(D) - D, AngularVelocityProbe1.phi(D) - D, DCmotor.EMF1.flange_b.phi(D) - D, DCmotor.Resistor1.t(t) - DCmotor.Resistor1.r(t), D, DCmotor.Inertia1.t(t) - DCmotor.Inertia1.phi(t) + DCmotor.Inertia1.flange_a.tau(t) + DCmotor.Inertia1.flange_b.tau(t), DCmotor.EMF1.r(t) - DCmotor.EMF1.p.r(t), DCmotor.EMF1.flange_b.tau(t) + IdealGear1.flange_a.tau(t) + IdealGear1.flange_b.tau(t) - $\frac{1}{5}$ Z1.u(t)

Exemple: 2,300 equations simplifiées à 150

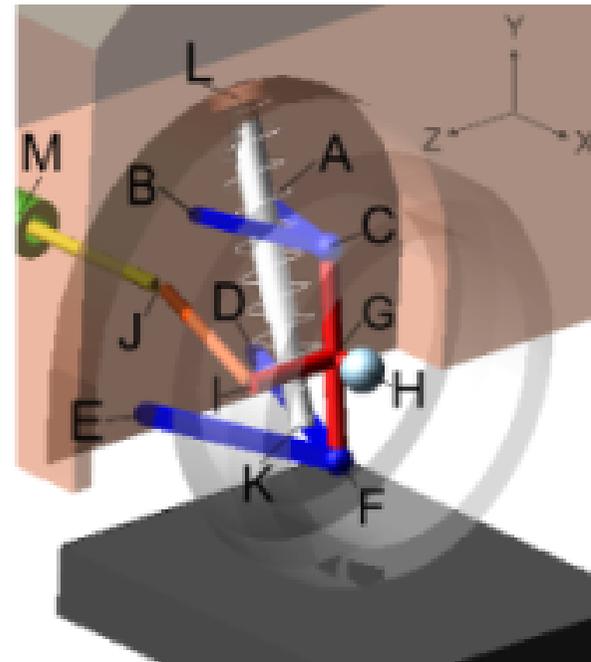
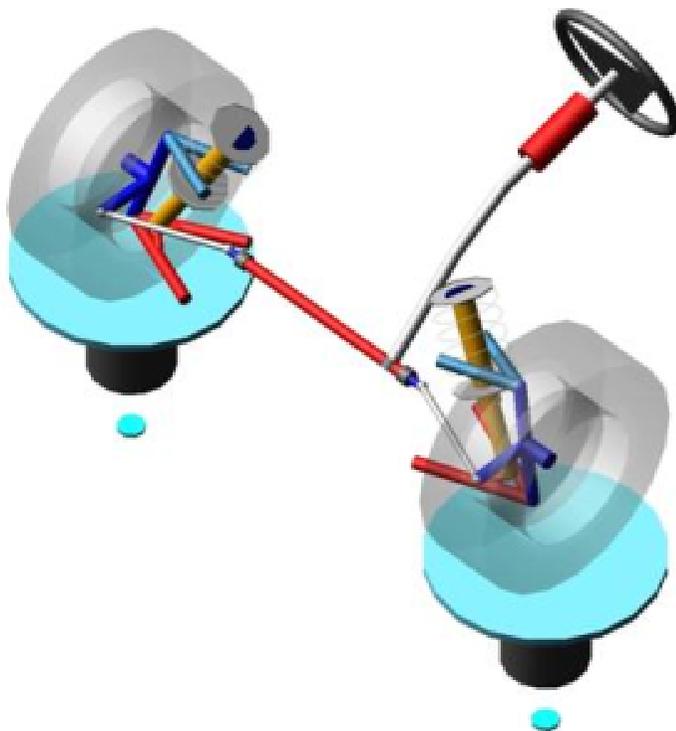
Accélération: 10 x

- Inertia2.phi(t) + AngularVelocityProbe2.phi(t) = 0, TorqueProbe1.flange_a.tau(t) + TorqueProbe2.flange_a.tau(t) = 0, DCmotor1.Inertia1.phi(t) - Inertia2.phi(t) + SpringDamper1.phi_rel(t) = 0, AngularVelocityProbe1.phi(t) + DCmotor1.Inertia1.phi(t) = 0, DCmotor1.EMF1.flange_b.phi(t) + DCmotor1.Inertia1.phi(t) = 0, Inertia2.w(t) - SpeedSensor1.flange_a.phi(t) = 0, TorqueProbe2.flange_a.tau(t) + SpringDamper1.phi_rel(t) + $\frac{d}{dt}$ SpringDamper1.phi_rel(t) = 0, DCmotor1.Inertia1.w(t) - $\left(\frac{d}{dt} DCmotor1.Inertia1.phi(t)\right) = 0, \frac{d}{dt} DCmotor1.Inductor1.i(t) + DCmotor1.Inductor1.i(t) - 10 \left(\frac{d}{dt} I1.x(t)\right) - 2.I1.x(t) + \frac{d}{dt} DCmotor1.EMF1.flange_b.phi(t) = 0, -10 \left(\frac{d}{dt} I1.x(t)\right) - 2 \left(\frac{d}{dt} SpeedSensor1.flange_a.phi(t)\right) + 2 \left(\begin{cases} 0, & t < 0 \\ 1, & otherwise \end{cases}\right) = 0, AngularVelocityProbe2.w(t) - \left(\frac{d}{dt} AngularVelocityProbe2.phi(t)\right) = 0, -AngularVelocityProbe1.w(t) + \frac{d}{dt} AngularVelocityProbe1.phi(t) = 0, -Inertia2.w(t) + \frac{d}{dt} Inertia2.phi(t) = 0, -TorqueProbe2.flange_a.tau(t) + \frac{d}{dt} Inertia2.w(t) = 0, -2.DCmotor1.Inductor1.i(t) + 2.TorqueProbe1.flange_a.tau(t) + \frac{d}{dt} DCmotor1.Inertia1.w(t) = 0, SpringDamper1.phi_rel(0) = 0, DCmotor1.Inertia1.w(0) = 0, I1.x(0) = 0, Inertia2.w(0) = 0, AngularVelocityProbe2.phi(0) = 0, DCmotor1.Inertia1.phi(0) = 0, Inertia2.phi(0) = 0, DCmotor1.Inductor1.i(0) = 0, SpeedSensor1.flange_a.phi(0) = 0, AngularVelocityProbe1.phi(0) = 0, DCmotor1.EMF1.flange_b.phi(0) = 0$

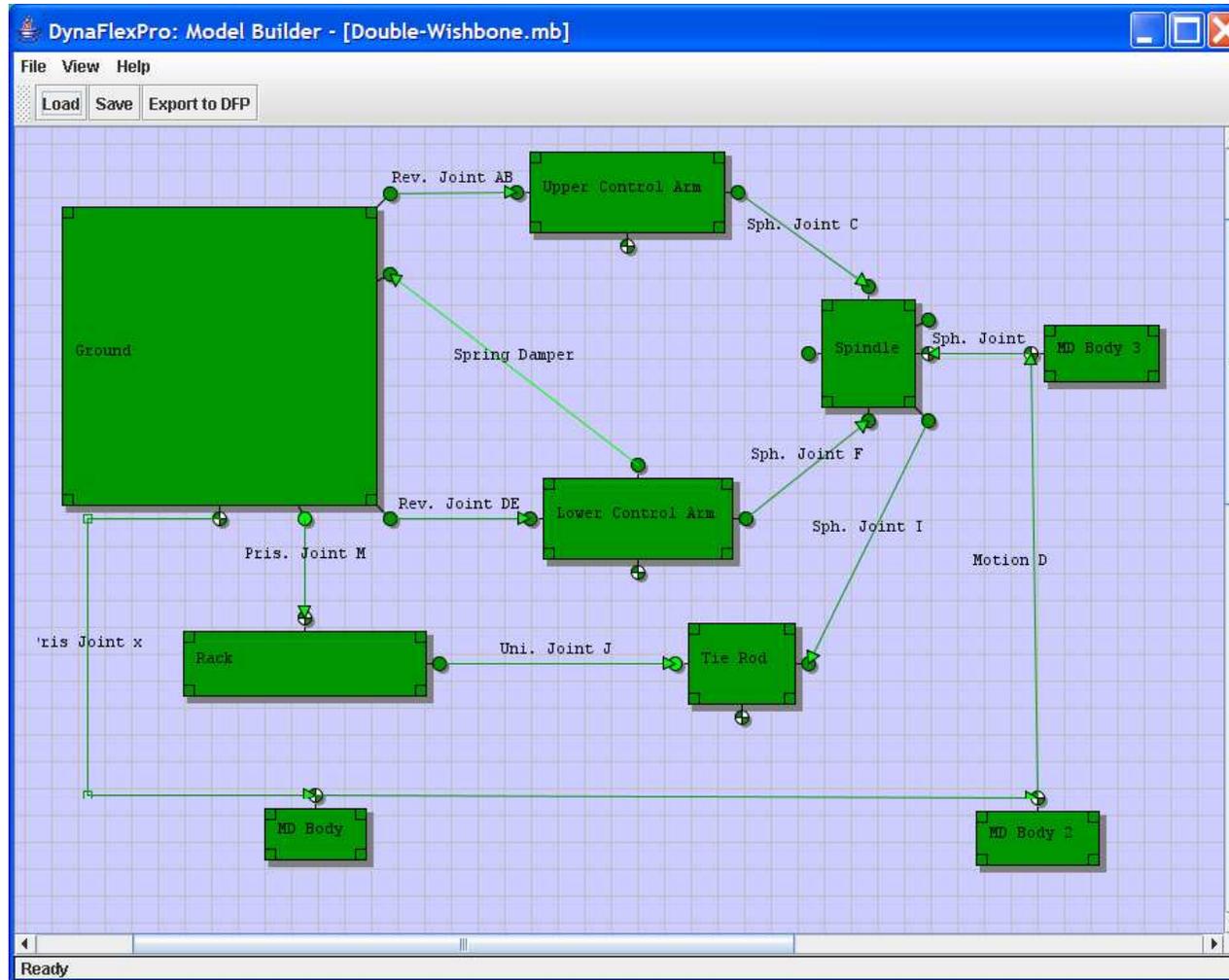


Exemples d'application et apport du calcul formel

Modèle inverse : optimisation d'une cinématique



Topologie du modèle



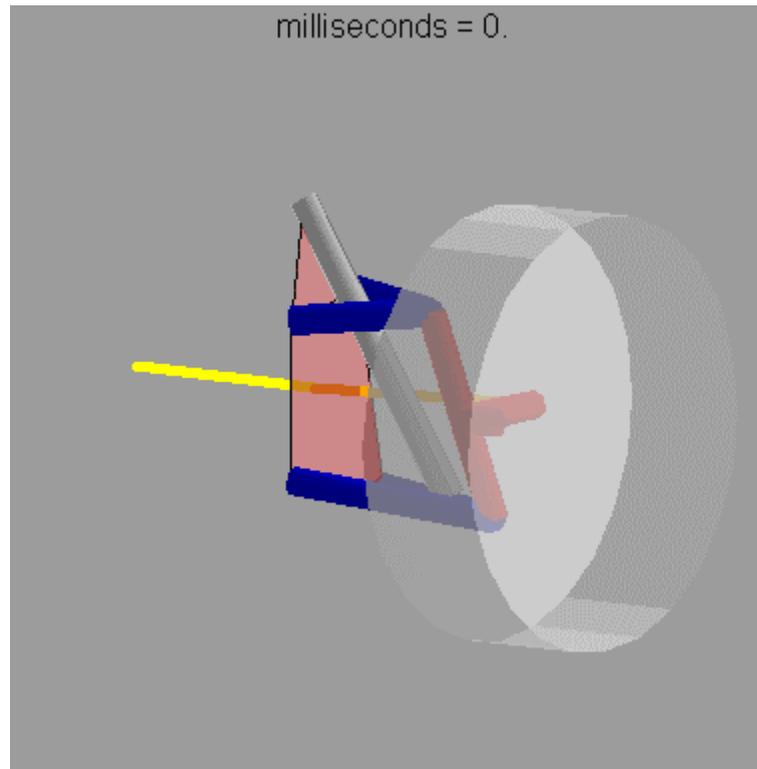
Equations du système

```

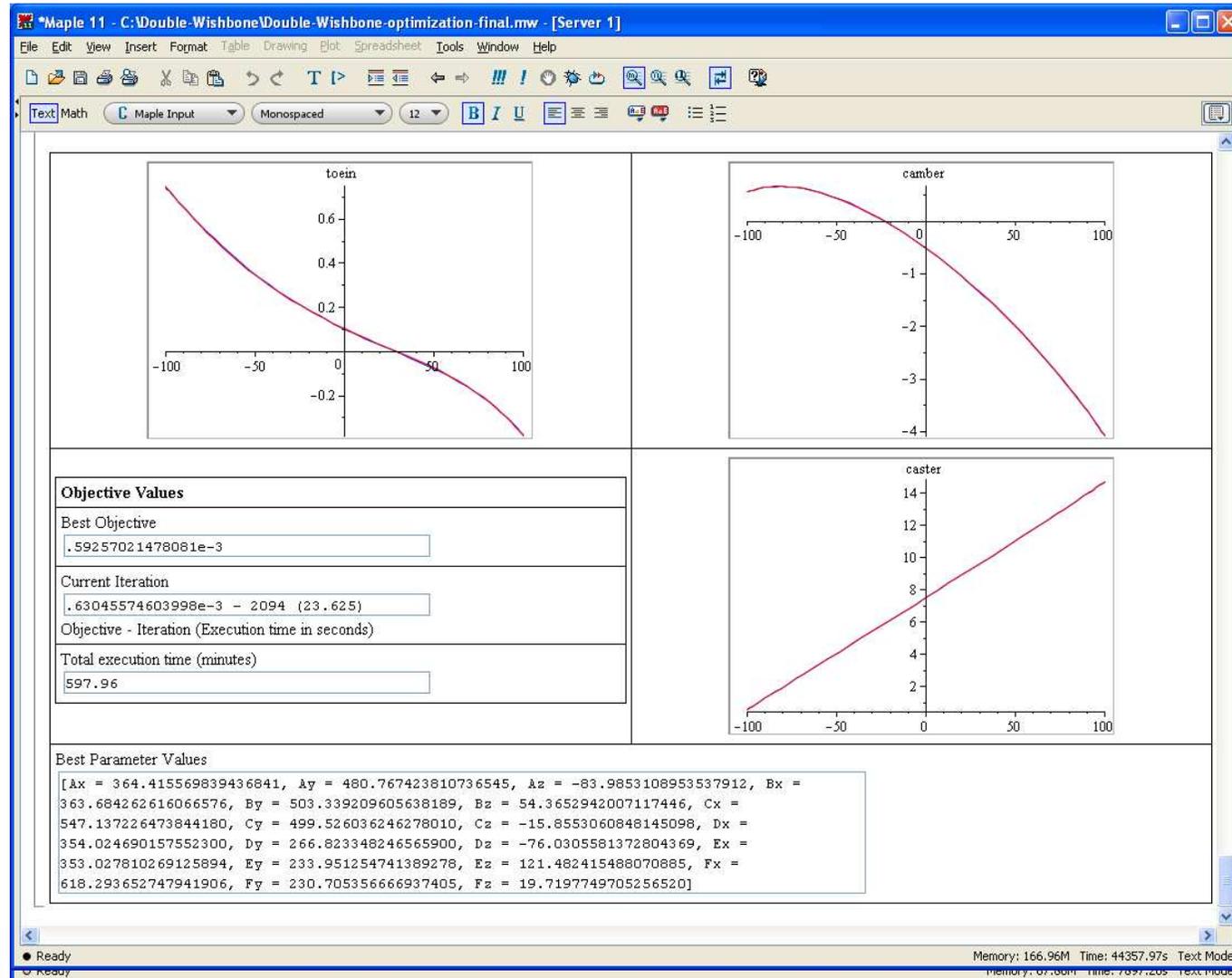
Maple 10 - C:\Documents and Settings\pgoossens\My Documents\Maplesoft\Customer Applications\MotionPro\White Paper\Double-Wishbone White Paper...
File Edit View Insert Format Table Plot Spreadsheet Sketch Tools Window Help
[Icons]
[Icons]
Text Math C Text Arial 10 B I U [Icons] [Icons] [Icons]
> UpperControlArm := [UCA_ABx = (Ax + Bx) / 2 - Cx, UCA_ABy = (Ay + By) / 2 - Cy, UCA_ABz = (Az + Bz) / 2 - Cz, UCA_Cx = 0, UCA_Cy = 0,
UCA_Cz = 0, AngAB_X = eval(arcsin((Ay - By) / sqrt((Ax - Bx)^2 + (Ay - By)^2 + (Az - Bz)^2))),
AngAB_Y = eval(arcsin(-(Ax - Bx) / sqrt((Ax - Bx)^2 + (Ay - By)^2 + (Az - Bz)^2)))] :
> LowerControlArm := [LCA_DEx = (Dx + Ex) / 2 - Fx, LCA_DEy = (Dy + Ey) / 2 - Fy, LCA_DEz = (Dz + Ez) / 2 - Fz, LCA_Fx = 0, LCA_Fy = 0,
LCA_Fz = 0, LCA_Kx = Ex - Fx, LCA_Ky = Ey - Fy, LCA_Kz = Ez - Fz, AngDE_X = eval(arcsin((Dy - Ey) / sqrt((Dx - Ex)^2 + (Dy - Ey)^2 + (Dz - Ez)^2))),
AngDE_Y = eval(arcsin(-(Dx - Ex) / sqrt((Dx - Ex)^2 + (Dy - Ey)^2 + (Dz - Ez)^2)))] :
#Parameter for the Spindle (also called the wheel knuckle, or wheel hub carrier)
> Spindle := [Sp_Cx = Cx - Hx, Sp_Cy = Cy - Hy, Sp_Cz = Cz - Hz, Sp_Fx = Fx - Hx, Sp_Fy = Fy - Hy, Sp_Fz = Fz - Hz, Sp_Ix = Ix - Hx, Sp_Iy = Iy - Hy,
Sp_Iz = Iz - Hz, Sp_Gx = Gx - Hx, Sp_Gy = Gy - Hy, Sp_Gz = Gz - Hz, WheelAngY = eval(arcsin(-(Hz - Gz) / sqrt((Hx - Gx)^2 + (Hy - Gy)^2 + (Hz - Gz)^2))),
WheelAngZ = eval(arcsin((Hy - Gy) / sqrt((Hx - Gx)^2 + (Hy - Gy)^2 + (Hz - Gz)^2)))] :
> TieRod := [TR_Ix = 0, TR_Iy = 0, TR_Iz = 0, TR_Jx = Jx - Ix, TR_Jy = Jy - Iy, TR_Jz = Jz - Iz,
AngleJ_Y = eval(arcsin(-(Iz - Jz) / sqrt((Ix - Jx)^2 + (Iy - Jy)^2 + (Iz - Jz)^2))), AngleJ_Z = eval(arcsin((Iy - Jy) / sqrt((Ix - Jx)^2 + (Iy - Jy)^2 + (Iz - Jz)^2)))] :
> Rack := [Ra_Jx = Jx - Mx, Ra_Jy = Jy - My, Ra_Jz = Jz - Mz] :
> Chassis := [Ch_ABx = (Ax + Bx) / 2, Ch_ABy = (Ay + By) / 2, Ch_ABz = (Az + Bz) / 2, Ch_DEx = (Dx + Ex) / 2, Ch_DEy = (Dy + Ey) / 2,
Ch_DEz = (Dz + Ez) / 2, Ch_Lx = Lx, Ch_Ly = Ly, Ch_Lz = Lz, Ch_Mx = Mx, Ch_My = My, Ch_Mz = Mz] :
> SpringDamper := [Ks = SpringK, Kd = SpringD, Kl = sqrt((Lx - Kx)^2 + (Ly - Ky)^2 + (Lz - Kz)^2)] :
> Params := [op(UpperControlArm), op(LowerControlArm), op(Spindle), op(TieRod), op(Rack), op(Chassis), op(SpringDamper)] :

```

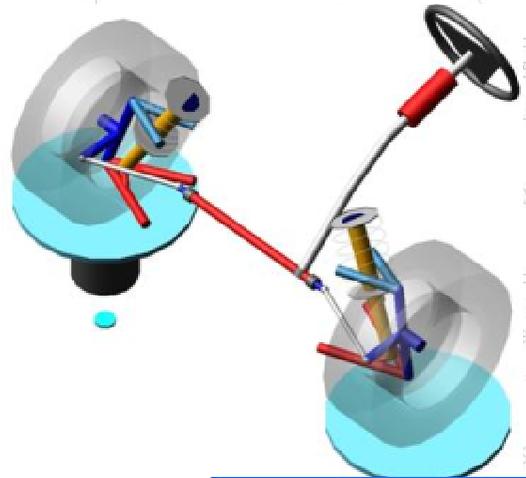
Modèle avec paramètres physiques pertinents



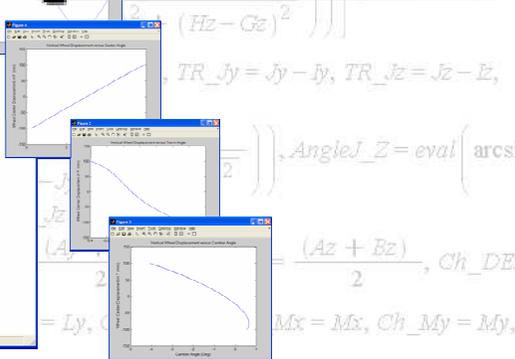
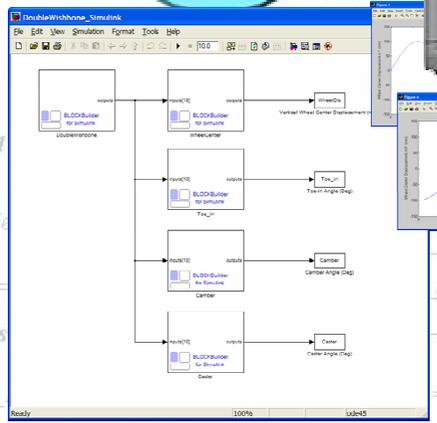
Optimization des paramètres



Intégration des composants physiques, Recallage sur données réelles



$UCA_Cz = 0, AngAB_X = eval\left(\arcsin\left(\frac{(Ay - By)}{\sqrt{(Ax - Bx)^2 + (Ay - By)^2 + (Az - Bz)^2}}\right)\right)$
 $\frac{(Bx)^2 + (Ay - By)^2 + (Az - Bz)^2}{(Ax - Bx)}$
 $\frac{Dx + Ex}{2} - Fx, LCA_DEy = \frac{(Dy + Ey)}{2} - Fy, LCA_DE$
 $TA_Ky = Ky$
 $Ex)^2 + ($
 $ael knuckel,$
 $= Cy - Hy$
 $p_Cy = C$
 $(Hz - Gz)^2$
 $TR_Jy = Jy - Jy, TR_Jz = Jz - Jz,$
 $AngleJ_Z = eval\left(\arcsin\left(\frac{(Az + Bz)}{2}\right)\right), Ch_DEX$
 $= Ly, C$
 $Mx = Mx, Ch_My = My,$
 $Ks = SpringK, Kd = SpringD, Kl = \sqrt{(Lx - Kx)^2 + (Ly - Ky)^2 + (Lz - Kz)^2}$
 $Params := [op(UperControlArm), op(LowerControlArm), op(Spindle), op(TieRod), op(Rack), op(Chassis), op(SpringDamper)]$

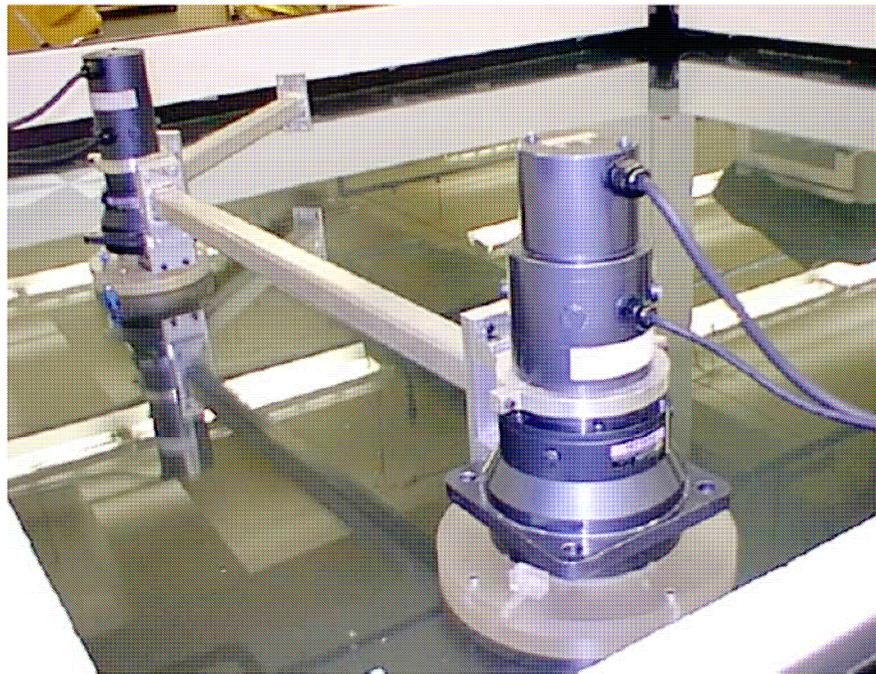


Exemples d'application et apport du calcul formel

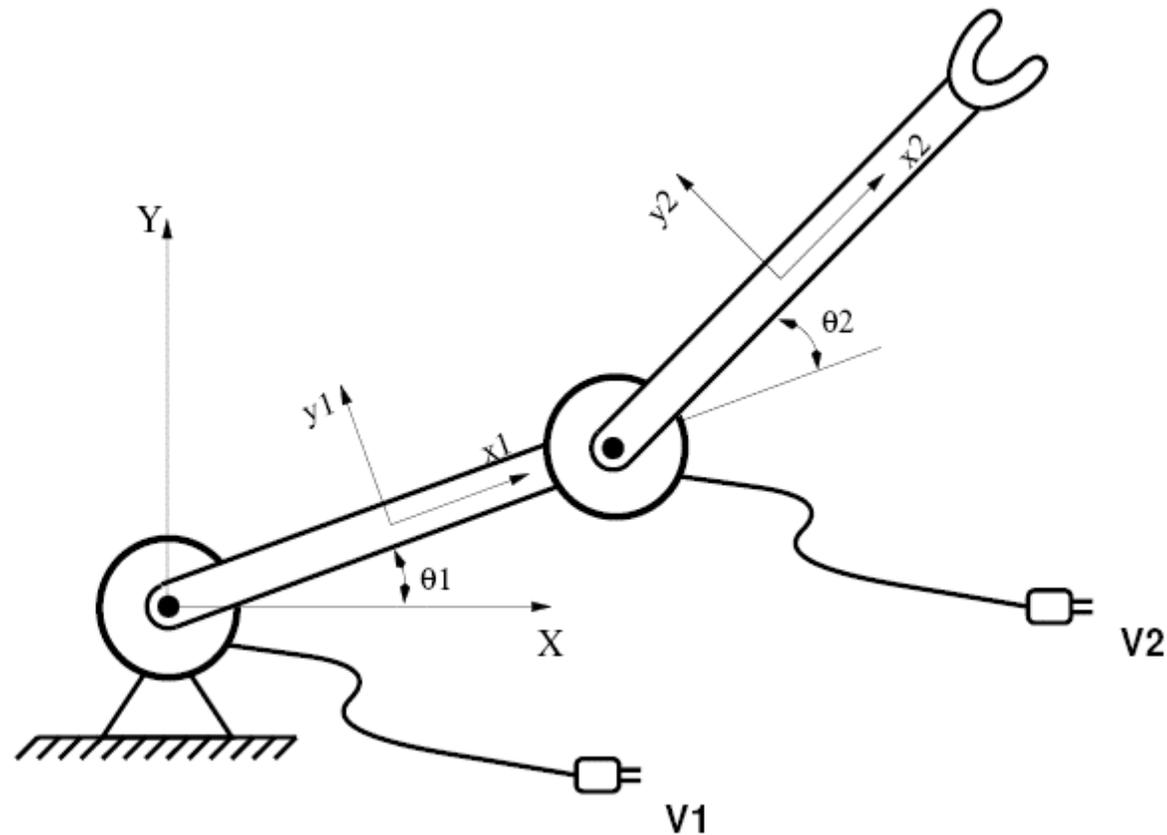
Systeme électromécanique :

Modélisation des modes souples

Termes de couplage dans le système d'équations



Modèle système



Moteurs électriques pour piloter les articulations

Termes couplés

Equations de chaque domaine =

Obtenues en utilisant les formulations électriques et mécaniques

Equations constitutives des transducteurs : couplées

$$v = K_v \frac{d\theta}{dt} + Ri + L \frac{di}{dt}$$

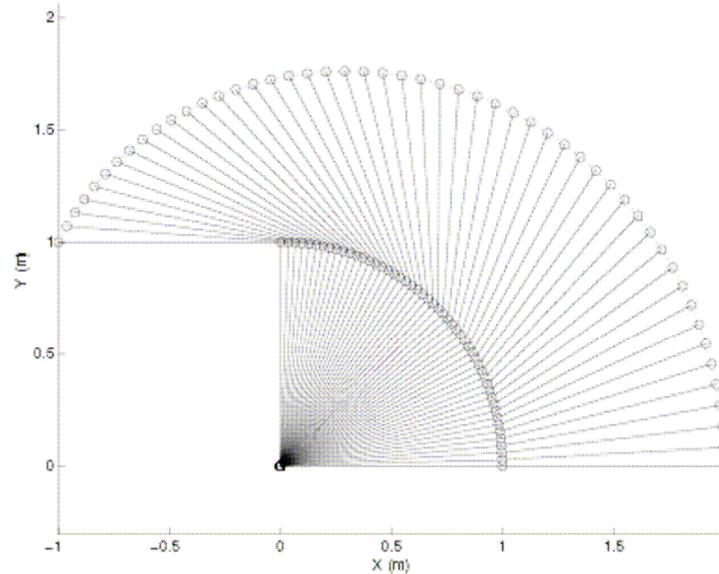
$$T = K_T i - B \frac{d\theta}{dt}$$

Avec le calcul symbolique =

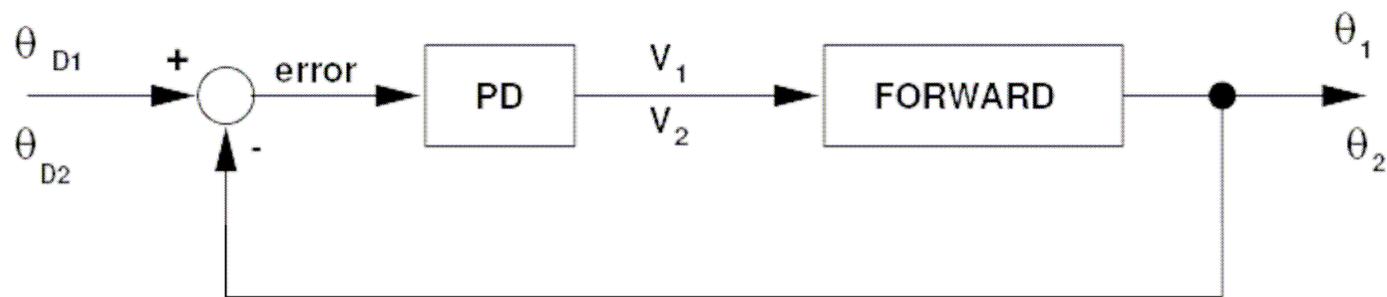
termes de couplage exprimés simplement

Utilisation directe de la formulation

Trajectoire souhaitée :

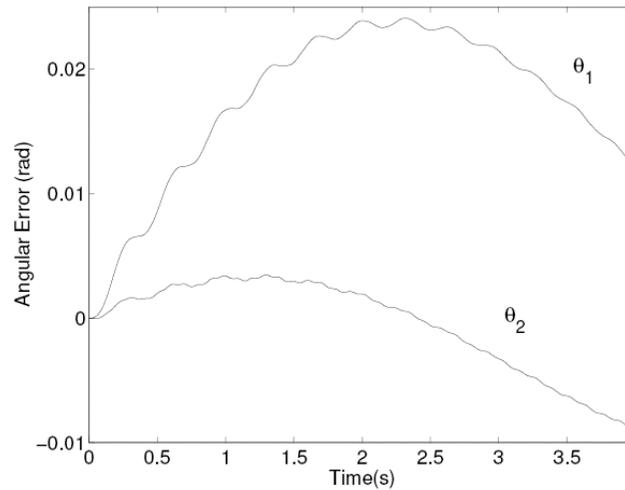


Boucle de contrôle :

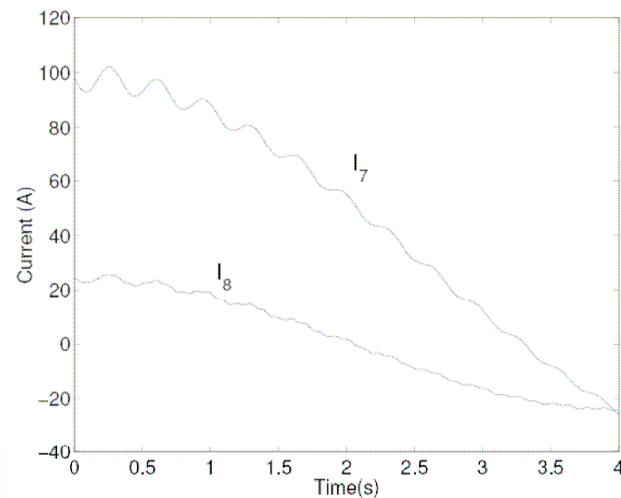


Utilisation directe de la formulation

Suivi trajectoire :

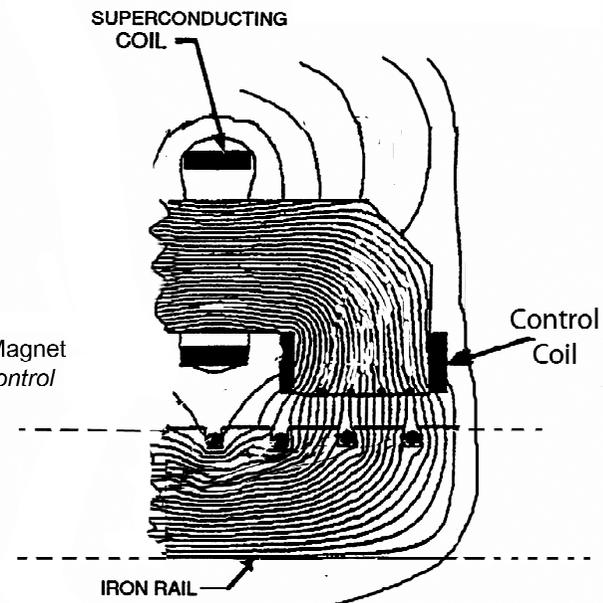
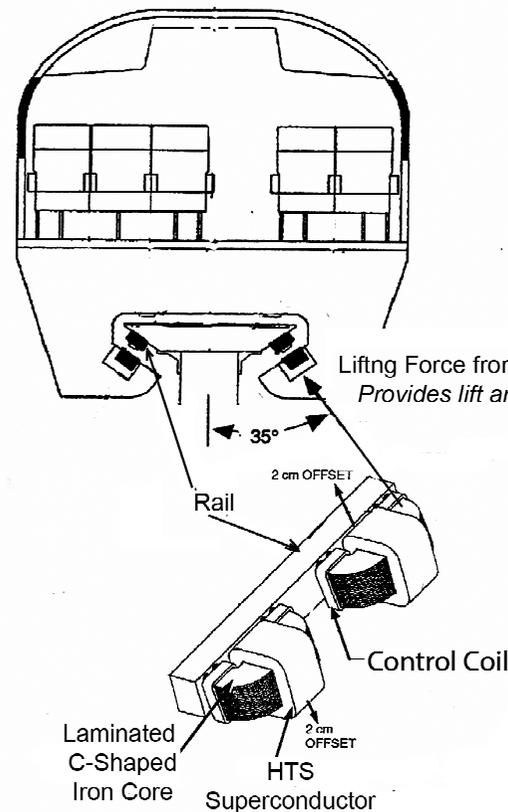


Courant dans les moteurs :



Exemples d'application et apport du calcul formel

MAGLEV



Cross Section of C-Shaped Magnet and Rail
Showing HTS Magnet and One of the 2 Control Coils
Magnetic Field Lines from a 2-D Calculation

[Document Maple](#)

Exemples d'application et apport du calcul formel

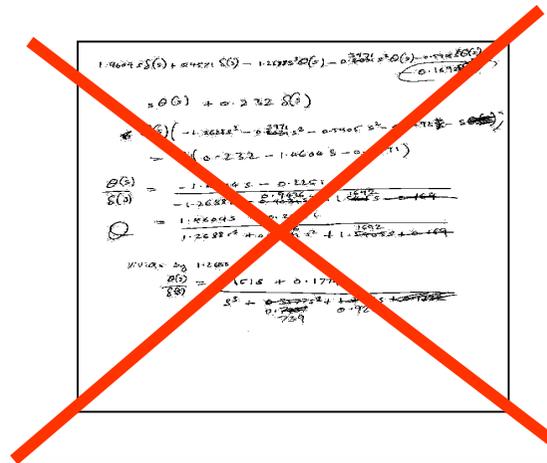
Vitesse de développement du modèle :

Bien plus rapide qu'une approche « traditionnelle »

(Richard Gran, ancien directeur concepts avancés chez Grumman Corp)

Formulation avec le papier et le crayon :

Erreurs, modification impossible, conservation limitée,...



Exemples d'application et apport du calcul formel

Avec le calcul formel =

Manipulations instantanées et conservation des équations sous forme modifiable

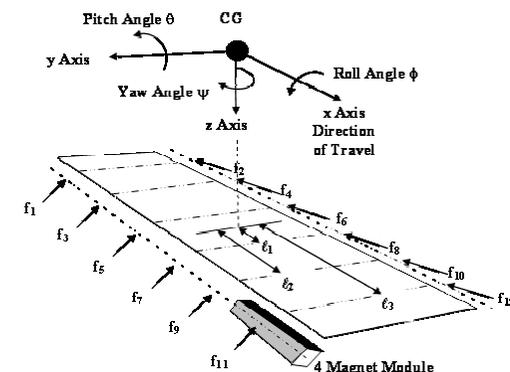
Modèle d'aimant :

$$ode1 := diff(v(t), t) + g = \frac{f(z(t), I_c(t))}{m} + \frac{Fd(t)}{m}$$

$$ode2 := diff(z(t), t) = v(t)$$

$$ode3 := diff(v(t), t) = \frac{Kz \cdot z(t)}{m} + \frac{KI \cdot I_c(t)}{m} + \frac{Fd(t)}{m}$$

Modèle multi-aimants :



Exemples d'application et apport du calcul formel

Manipulations automatiques =
modifications instantanées de la formulation

Meilleure interprétation physique des résultats

Utilisation directe du modèle pour la conception de correcteurs

Modèle écrit sous forme d'équations =
documenté naturellement (Connaissance sur le modèle et les choix de modélisation)

Avec les outils professionnels :
Solveurs numériques directement sur la formulation
Connexion directe aux outils de simulation

Exemples d'application et apport du calcul formel

Validation de modèle =
unités physiques, comparaison, procédures de test

Dérivée formelle par rapport aux paramètres =
optimisation du plan d'expérience

Objets formels =
équations, procédures, modèles schéma bloc,...

Les nouvelles techniques

Document mathématique interactif

- Ecriture naturelle des équations sans syntaxe
- Commentaires et explications interactives
- Dessin et annotation intuitives
- Résultat : définition propre du modèle, pour qu'il puisse être conservé même sous forme imprimée
- Réutilisation simple et transmission efficace
- Lien avec code très simple
- Manipulation interactives du modèle
- Publication du modèle

Palettes adaptables

Notation Math naturelle

Modèles et assistants

Annotations

Opérations auto-documentées

Courbes annotées

Composants d'interface simples d'utilisation

Partage des résultats

Connectivité données

Circular Heat Sink (Irregular Loading)

Analytical Model

The model assumes a 2D disk and uses the Laplace heat equation for polar coordinates

$$\frac{\partial^2}{\partial r^2} K(r, \theta) + \frac{1}{r} \frac{\partial}{\partial r} K(r, \theta) + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2} K(r, \theta) = 0.$$

Where r and θ define the position on the disk. The variable $K(r, \theta)$ is temperature in Kelvins. The solution is a Fourier series expansion. For 5 terms, it is,

$$-\frac{1}{2a^2} r^2 \sin(2\theta) - \frac{4}{\pi} \sum_{k=1}^5 \frac{r^{2k-1}}{(2k+1)(2k-3)\pi a^{2k-1}} \cos(2k-1)\theta$$

$$-\frac{1}{6930} \frac{1}{\pi^2 a^6} (r(-3465r \sin(2\theta) \pi^2 a^7 + 9240\theta \cos(1) a^8 + 5544\theta r^2 \cos(3) a^6 + 3960\theta r^4 \cos(5) a^4 + 3080\theta r^6 \cos(7) a^2 + 2520\theta r^8 \cos(9)))$$

Validation

θ [[rad]]	r [[cm]]	$K(\theta, r)$ [[K]]
85.00×10^{-3}	3.42	-307.99×10^{-3}
140.00×10^{-3}	2.85	166.14×10^{-3}
183.00×10^{-3}	1.51	-250.23×10^{-3}

*Normalized at 300 Kelvins.

Validation: Model Solution vs. Lab Data

2D sections for values of θ

Comparison of solution accuracy for different series orders
 $N = 1, 2, 5, 20$
 $\theta = \frac{\pi}{40} = 78.54 \times 10^{-3}$

2D slice on disk (θ)

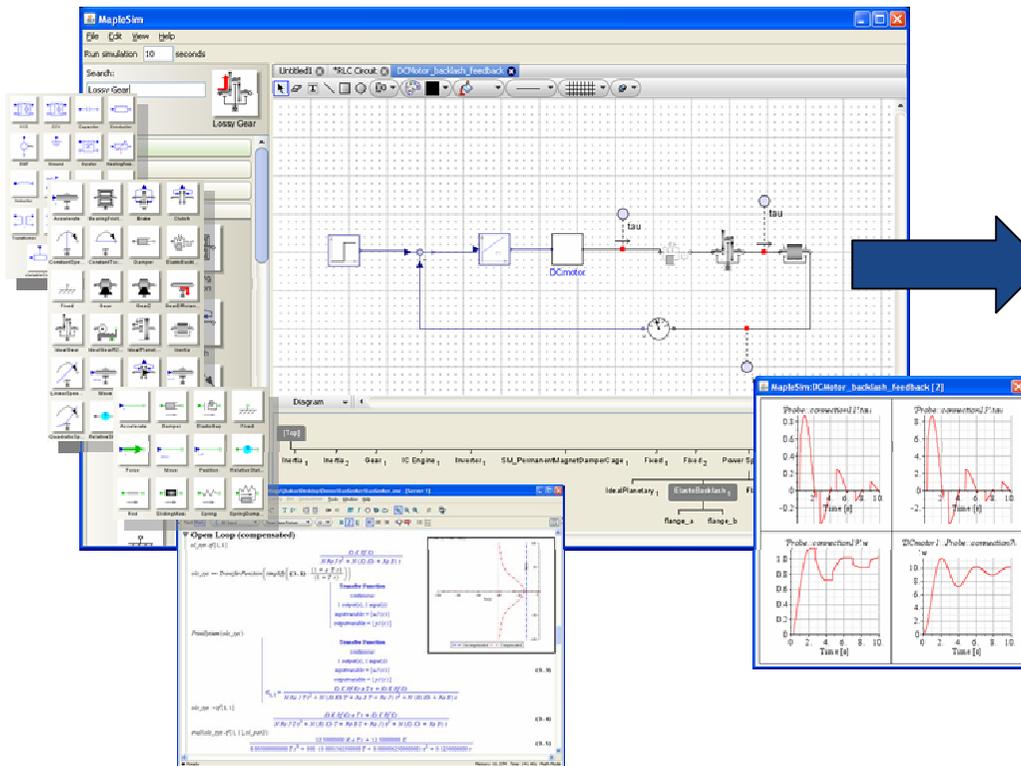
Les nouvelles techniques

Génération automatique d'équations

- Spécification du modèle sous forme intuitive
- Visualisation des équations générées pour leur analyse
- Gain de temps pour la production du modèle
- Utilisable par un grand nombre d'ingénieurs

Les nouvelles techniques

Génération automatique d'équations :



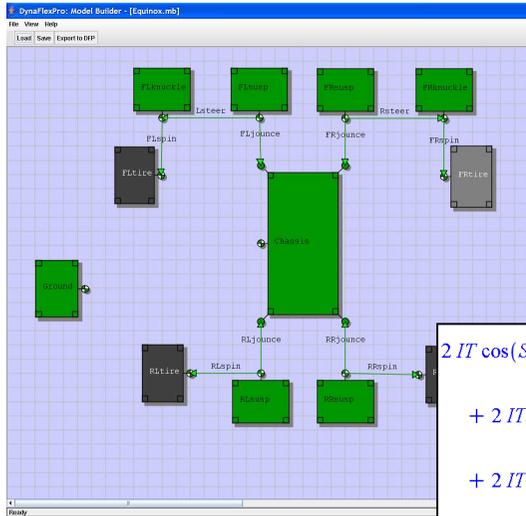
$$L \left(\frac{d^2}{dt^2} q(t) \right) + R \left(\frac{d}{dt} q(t) \right) + \frac{q(t)}{C} = v(t)$$

Les nouvelles techniques

Production de code optimisé pour le HIL

- Simplification formelle préalable
- Modèle proche de la physique
- Optimisation du nombre d'opérations
- Recalage direct des valeurs physiques avec mesures et modèle analytique

Génération de code : exemple

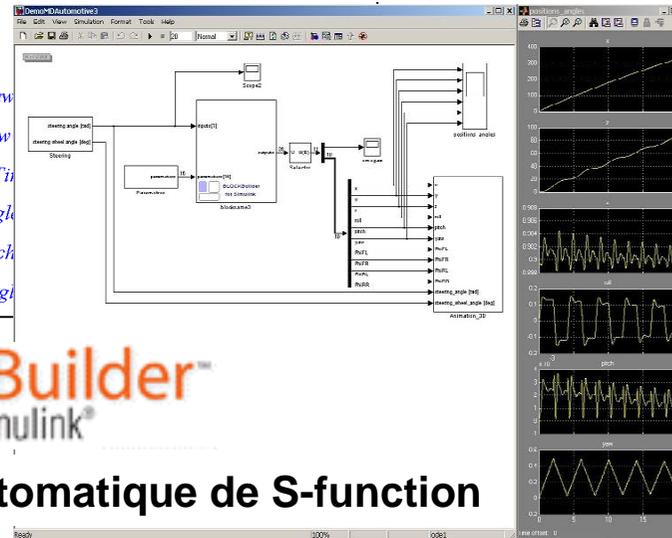


**Modèle dynamique véhicule
- 14 DoF plus 8 DoF Pneus**

Maple 11

Analyse/Validation/Documentation
Génération Automatique Equations
Cinématiques et Dynamiques

$$\begin{aligned}
 & 2 IT \cos(SteerAngle(t)) \left(\frac{d}{dt} SteerAngle(t) \right) \omega x(t) \\
 & + 2 IT \cos(SteerAngle(t)) \left(\frac{d}{dt} \omega y(t) \right) \\
 & + 2 IT \sin(SteerAngle(t)) \left(\frac{d}{dt} \omega x(t) \right) \\
 & - 2 \sin(SteerAngle(t)) \omega y(t) IT \left(\frac{d}{dt} SteerAngle(t) \right) \\
 & + 2 IT \left(\frac{d^2}{dt^2} \phi_{RR}(t) \right) \\
 & - \cos(SteerAngle(t)) \cos(roll(t)) \cos(yaw) \\
 & - \cos(SteerAngle(t)) \cos(roll(t)) \sin(yaw) \\
 & + \sin(pitch(t)) \sin(SteerAngle(t)) R1_Ti \\
 & - \cos(pitch(t)) \sin(yaw(t)) \sin(SteerAngle) \\
 & - \cos(SteerAngle(t)) \sin(roll(t)) \sin(pitch) \\
 & - \cos(pitch(t)) \cos(yaw(t)) \sin(SteerAngi
 \end{aligned}$$



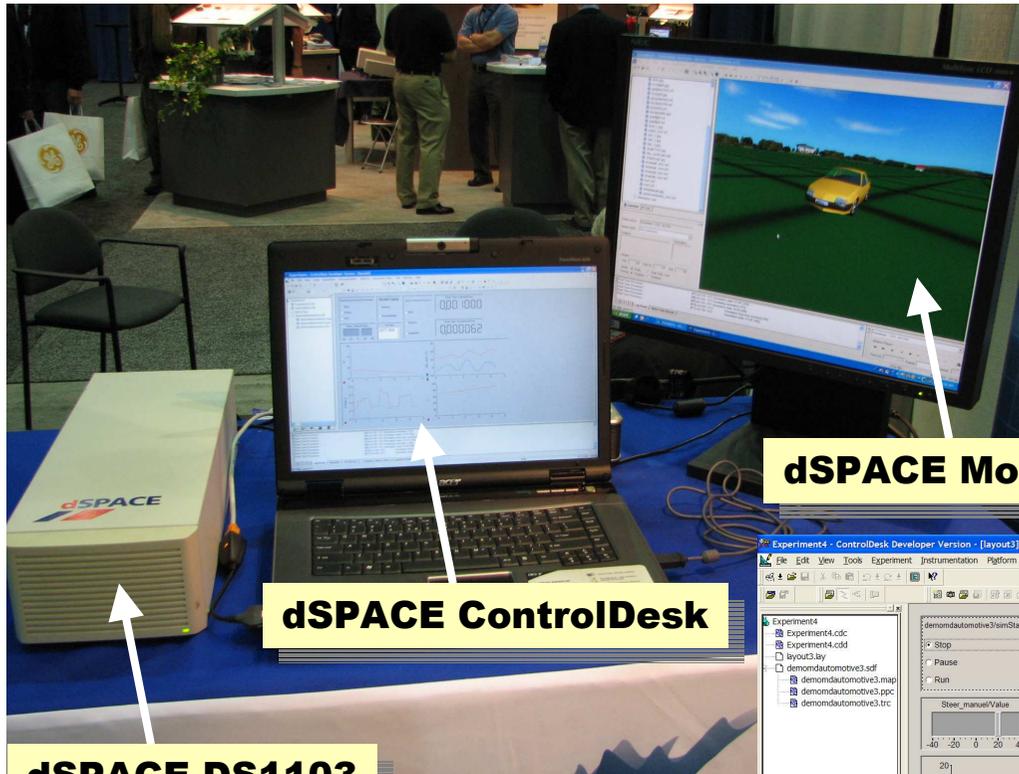
**Real-Time
Workshop**

BLOCKBuilder™
for Simulink®

Génération automatique de S-fonction



Génération de code : exemple

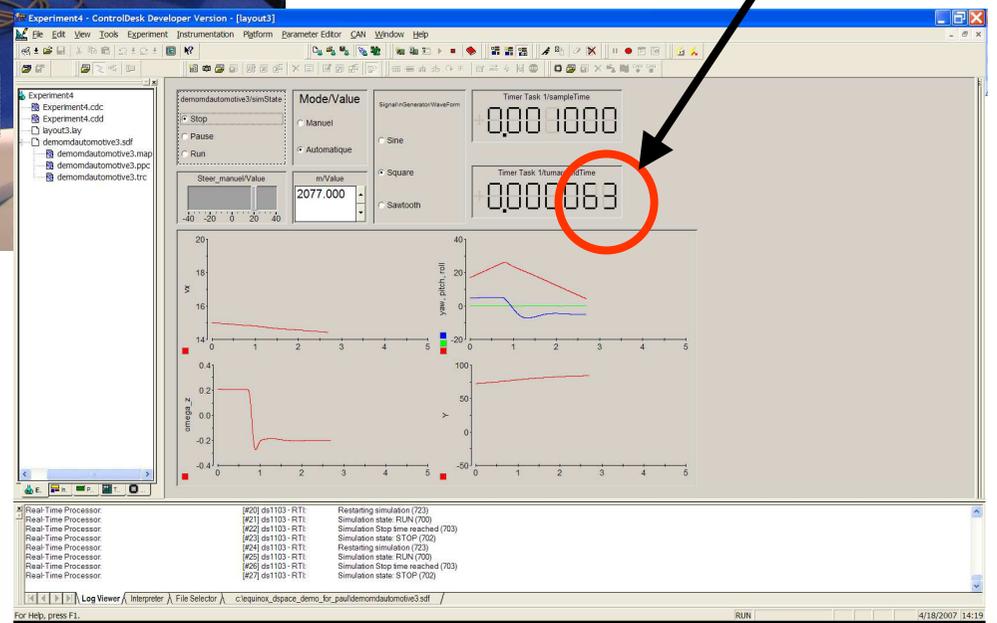


dSPACE DS1103

dSPACE ControlDesk

dSPACE MotionDesk

Temps de cycle 63µs!



Consortium PMC

Depuis Août 2007

Deux réunions, troisième prévue en octobre 08

www.pmconsortium.org

Démonstration

Modélisation et pilotage d'un avion

Modélisation et pilotage d'un système électromécanique

Conclusions

Calcul formel =

- Formulation automatique
- Génération de code optimisé
- Dérivée formelle (y compris sur procédures et schémas blocs)
- Optimisation de plan d'expérience
- Validation de modèle (y compris dimensions physiques)

Conclusions

Nouvelles techniques =

- Document Interactif :
 - Partage de la connaissance sur le modèle
 - Capitalisation du modèle
- Génération automatique d'équations
 - Utilisation par un plus grand nombre
 - Automatisation
 - Intégration aux « process » de développement



Questions?