Préservation de preuve lors de la compilation sur microcontrôleur

Vivien Maisonneuve

Préservation de preuve lors de la compilation sur microcontrôleur

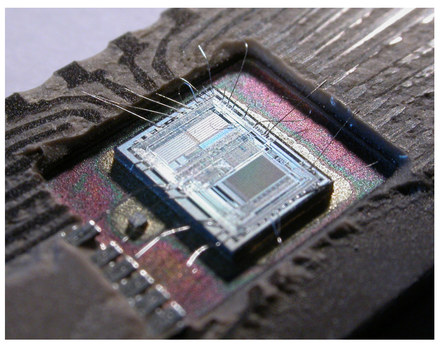
Vivien Maisonneuve, thèse encadrée par François Irigoin et Olivier Hermant
Centre de recherche en informatique, Mathématiques et systèmes, MINES ParisTech, Fontainebleau

Conception d'un système embarqué

Un **système embarqué** est un système informatique autonome spécialisé.
Contraintes d'autonomie, de temps d'exécution, de sécurité & sûreté.

Utilise un microprocesseur basse consommation ou un microcontrôleur.

Systèmes embarqués dans les moteurs, les télécommandes, les appareils de bureau, l'électroménager, les jouets, les téléphones, etc.



Travail de conception sur deux niveaux :

Formalisation :
- Conception du système ;
- Modélisation physique de l'environnement ;
- **Preuve** mathématique que le système se comporte correctement.

MATLAB, Simulink

Réalisation : **programme** C de très bas niveau
- Plusieurs milliers de LOC ;
- Calculs décomposés en opérations élémentaires ;
- Gestion des moteurs et des capteurs.

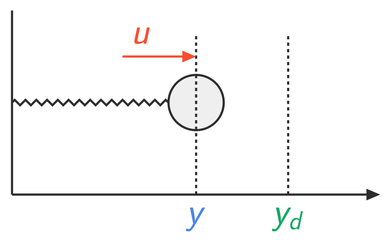
GCC, Clang

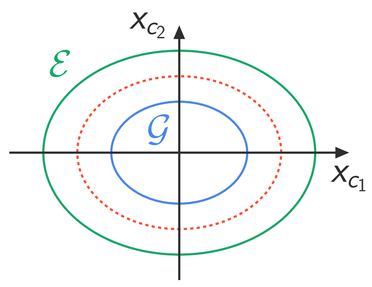
Transformations graduelles

Exemple : système masse-ressort [1]

Système mécanique à 1 degré de liberté : **masse** accrochée à un **ressort**, contrainte de se déplacer dans une seule direction.

Le contrôleur exerce une action u afin que la position y de la masse atteigne la consigne y_d.






```
Ac = [0.4990, -0.0500; 0.0100, 1.00];
Bc = [1; 0];
Cc = [564.48, 0];
Dc = -1280;
xc = zeros(2, 1);
receive(y, 2); receive(yd, 3);
while (1)
  % xc ∈ ℰ
  yc = max(min(y - yd, 1), -1);
  u = Cc*xc + Dc*yc;
  xc = Ac*xc + Bc*yc;
  send(u, 1);
  receive(y, 2);
  receive(yd, 3);
  % xc ∈ 𝒢 ⊂ ℰ
end
```

Condition de stabilité (Lyapunov) : la variable d'état x_c reste dans une certaine ellipse $\mathcal{E}$ durant l'exécution.

Preuve de stabilité dans R : fournie sous forme d'**invariants**
- À l'entrée de la boucle, x_c appartient à $\mathcal{E}$;
- En sortie, x_c appartient à une ellipse $\mathcal{G} \subset \mathcal{E}$;

En flottants ?
- Altération des constantes numériques A_c, B_c, C_c, D_c ;
- Erreurs d'arrondi lors du calcul de x_c.

$\Longrightarrow$ **Preuve inutilisable en flottants.**

Objectifs

Comment être sûr que le programme exécuté est correct ?

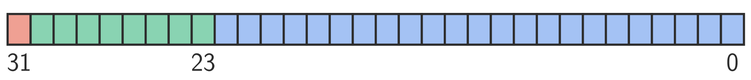
Preuves de **stabilité numérique** : montrer que les paramètres d'un système restent dans une certaine enveloppe durant son exécution (**stabilité de Lyapunov**).
Primordiales pour la sécurité du système.

Modélisation en **boucle ouverte** ou en **boucle fermée** (en tenant compte de la **rétroaction**).

Comment adapter la preuve sur le modèle physique en une preuve équivalente sur le programme ?

Nombres à virgule flottante (norme IEEE 754)

Dans le programme, les valeurs numériques du modèle sont **approximées** par des valeurs binaires de précision limitée, p. ex. **nombres flottants**.



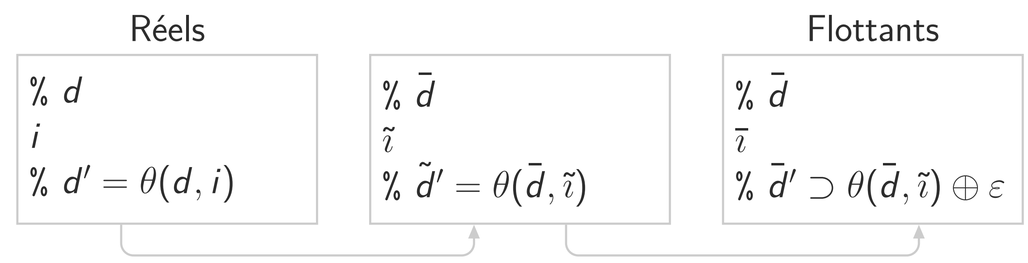
$(-1)^s \times 2^{e-127} \times m$

Les preuves de stabilité ne s'appliquent plus :
- Altération des constantes numériques du système ;
- **Erreurs d'arrondi** lors des calculs, qui se cumulent.

Approche pour prouver la stabilité en flottants :
- **Transposition** des arguments de preuve en tenant compte de ces effets ;
- Vérification de la validité de la preuve : si oui, le programme est stable.

Passage aux flottants – cadre théorique

Transposition code + invariants en 2 étapes :



Code : **constantes** converties en flottants

Invariants recalculés en utilisant les mêmes théorèmes θ, appliqués aux nouvelles constantes

Code : **fonctions** (+, ∗, . . .) remplacées par leurs équivalents flottants
Invariants « élargis » pour inclure l'erreur d'arrondi
Conservation de la forme ellipsoïdale pour propagation

Cadre prouvé en Coq.

Passage aux flottants – automatisation

LyaFloat : implémentation pour systèmes linéaires à invariants de Lyapunov
- Traduction du programme en flottants ;
- Transposition **automatique** des invariants ;
- Vérification de la condition de stabilité.

Possibilité de jouer sur la **précision**.

Programmé en Python ($\sim$ 500 LOC) avec la bibliothèque SymPy.

Bibliographie

[1] E. Feron. From Control Systems to Control Software. *IEEE Control Systems Magazine* 30(6):50–71, Dec. 2010.
[2] J. Feret. Static Analysis of Digital Filters. ESOP 2004, LNCS 2986, Springer (2004) 33–48.

Pour en savoir plus :
`http://www.cri.ensmp.fr/classement/doc/A-556.pdf`

Application au système masse-ressort

Calcul d'une ellipse $\mathcal{F}$ (en pointillés rouges ci-dessus) déduite de $\mathcal{G}$ telle qu'en sortie de boucle $x_c \in \mathcal{F}$, puis vérification de l'inclusion $\mathcal{F} \subset \mathcal{E}$.

Résultats :
- **Boucle ouverte** : système stable avec des flottants 32 bits.
- **Boucle fermée** : échec de l'analyse avec 32 bits, fonctionne avec 128 bits.