

# Formalisation de la stabilité de l'erreur d'estimation de l'inclinaison d'un robot

Lynda Bentoucha<sup>1</sup> Reynald Affeldt<sup>2</sup>

<sup>1</sup>Université Paris Cité

<sup>2</sup>Nat. Inst. of Advanced Industrial Science and Technology, Tokyo, Japon

Journées Francophones des Langages Applicatifs,  
Janvier 2026

# Plan

- 1 Introduction
- 2 L'estimation et sa formalisation
- 3 Formalisation de la stabilité
- 4 Application
- 5 Conclusion

# Caractéristiques d'un robot humanoïde

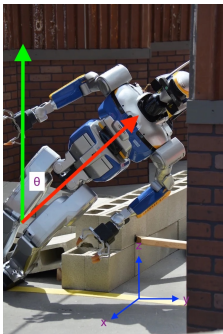
	Taille	Poids	Prix
HRP-2	1,54 m	58 kg	400 000€



Figure: HRP-2, développé par le CNRS-AIST JRL



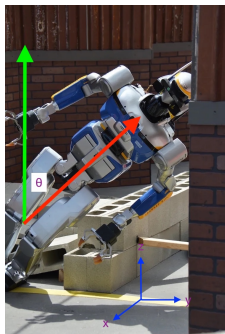
# Problème de l'équilibre d'un robot humanoïde



## “Tilt”

Inclinaison (“roll” et “pitch”) du  
champ de gravité du monde  
par rapport à  
l'axe vertical du robot

# Problème de l'équilibre d'un robot humanoïde



## “Tilt”

Inclinaison (“roll” et “pitch”) du  
champ de gravité du monde  
par rapport à  
l'axe vertical du robot

Son estimation précise permet de :

- détecter une perte d'équilibre
- corriger dynamiquement la trajectoire

# Instruments pour estimer le tilt

Pas de capteur dédié pour le tilt  
⇒ on utilise une combinaison d'instruments :

# Instruments pour estimer le tilt

Pas de capteur dédié pour le tilt

⇒ on utilise une combinaison d'instruments :

- ① IMU (“centrale inertielle” en français)
  - aussi dans les montres connectées, les drones, la Wiimote, etc.

composée de :

- Accéléromètre
- Gyroscope (vitesse de rotation)

# Instruments pour estimer le tilt

Pas de capteur dédié pour le tilt

⇒ on utilise une combinaison d'instruments :

- ① IMU (“centrale inertielle” en français)
  - aussi dans les montres connectées, les drones, la Wiimote, etc.composée de :
  - Accéléromètre
  - Gyroscope (vitesse de rotation)
- ② Capteur de vitesse
  - En pratique, l'IMU seule ne suffit pas pour un robot en mouvement

# Approche en robotique [Benallegue et al., 2023]

Pour l'estimation à l'aide de capteurs :

- 1 Obtenir une équation à propos du tilt et de son estimation
- 2 Modéliser une équation différentielle

$$f'(t) = \phi(t, f(t))$$

qui capture la dynamique des erreurs d'estimation

- 3 Montrer que l'erreur d'estimation converge vers 0
  - Ce qui montre que l'estimation est correcte

# Cette présentation

On entreprend de formaliser [Benallegue et al., 2023].

Pourquoi ?

- Motivation pour un roboticien:  
création de standards modernes pour la robotique
- Motivation pour ROCQ: enrichir les bibliothèques
  - ROBOTROCQ: géométrie 3D pour bras robotisés [Affeldt and Cohen, 2017]
  - MATHCOMP-ANALYSIS: analyse mathématique [Affeldt et al., 2018, Affeldt et al., 2020, Affeldt and Stone, 2024]

# Plan

- 1 Introduction
- 2 L'estimation et sa formalisation
- 3 Formalisation de la stabilité
- 4 Application
- 5 Conclusion

# Modélisation mathématique du problème physique (1/2)

- Repères monde et local :  $W = (e_x, e_y, e_z)$        $L = (\vec{i}, \vec{j}, \vec{k})$

# Modélisation mathématique du problème physique (1/2)

- Repères monde et local :  $W = (e_x, e_y, e_z)$        $L = (\vec{i}, \vec{j}, \vec{k})$
- Matrice de passage de  $L$  à  $W$ :  $R_{L/W} = \begin{pmatrix} \vec{i}_x & \vec{i}_y & \vec{i}_z \\ \vec{j}_x & \vec{j}_y & \vec{j}_z \\ \vec{k}_x & \vec{k}_y & \vec{k}_z \end{pmatrix}$
- Que mesure l'accéléromètre de l'IMU ?  $\rightarrow$  une valeur  $y_a$

# Modélisation mathématique du problème physique (1/2)

- Repères monde et local :  $W = (e_x, e_y, e_z)$       $L = (\vec{i}, \vec{j}, \vec{k})$
- Matrice de passage de  $L$  à  $W$ :  $R_{L/W} = \begin{pmatrix} \vec{i}_x & \vec{i}_y & \vec{i}_z \\ \vec{j}_x & \vec{j}_y & \vec{j}_z \\ \vec{k}_x & \vec{k}_y & \vec{k}_z \end{pmatrix}$
- Que mesure l'accéléromètre de l'IMU ? → une valeur  $y_a$   
Que représente  $y_a$  ?

# Modélisation mathématique du problème physique (1/2)

- Repères monde et local :  $W = (e_x, e_y, e_z)$       $L = (\vec{i}, \vec{j}, \vec{k})$
- Matrice de passage de  $L$  à  $W$ :  $R_{L/W} = \begin{pmatrix} \vec{i}_x & \vec{i}_y & \vec{i}_z \\ \vec{j}_x & \vec{j}_y & \vec{j}_z \\ \vec{k}_x & \vec{k}_y & \vec{k}_z \end{pmatrix}$
- Que mesure l'accéléromètre de l'IMU ? → une valeur  $y_a$   
Que représente  $y_a$  ? → accélération linéaire + gravitationnelle

# Modélisation mathématique du problème physique (1/2)

- Repères monde et local :  $W = (e_x, e_y, e_z)$        $L = (\vec{i}, \vec{j}, \vec{k})$
- Matrice de passage de  $L$  à  $W$ :  $R_{L/W} = \begin{pmatrix} \vec{i}_x & \vec{i}_y & \vec{i}_z \\ \vec{j}_x & \vec{j}_y & \vec{j}_z \\ \vec{k}_x & \vec{k}_y & \vec{k}_z \end{pmatrix}$
- Que mesure l'accéléromètre de l'IMU ? → une valeur  $y_a$   
Que représente  $y_a$  ? → accélération linéaire + gravitationnelle  
Supposons donnée la position  $p$ :

# Modélisation mathématique du problème physique (1/2)

- Repères monde et local :  $W = (e_x, e_y, e_z)$       $L = (\vec{i}, \vec{j}, \vec{k})$
- Matrice de passage de  $L$  à  $W$ :  $R_{L/W} = \begin{pmatrix} \vec{i}_x & \vec{i}_y & \vec{i}_z \\ \vec{j}_x & \vec{j}_y & \vec{j}_z \\ \vec{k}_x & \vec{k}_y & \vec{k}_z \end{pmatrix}$
- Que mesure l'accéléromètre de l'IMU ? → une valeur  $y_a$   
Que représente  $y_a$  ? → accélération linéaire + gravitationnelle  
Supposons donnée la position  $p$ :

$$y_a = (\ddot{p} + g_0 e_z) R_{L/W}^T$$

En ROCQ:

**Lemma**  $y_a \in \mathbb{R}^3$  :

$$y_a = \mathcal{D}_1 (\mathcal{D}_1 p) + g_0 * e_2 \in \mathbb{R}^3.$$

## Modélisation mathématique du problème physique (2/2)

Que mesure le gyroscope de l'IMU ? → une valeur  $\omega$

Or, pour une matrice de rotation  $R(t)$ , on a:

$$\frac{dR(t)}{dt} = R(t)S(\omega(t))$$

avec

$$S(\omega) = \begin{pmatrix} 0 & \omega_x & -\omega_y \\ -\omega_z & 0 & \omega_x \\ \omega_y & -\omega_x & 0 \end{pmatrix}$$

D'où le modèle physique (avec  $v$ ,  $\omega$ ,  $y_a$  seulement):

$$\begin{aligned} v \\ \omega \\ y_a \end{aligned} = -vS(\omega) + \dot{v} + g_0 e_z R^T.$$

# Obtenir une équation diff. qui fait apparaître le tilt

On considère un vecteur  $x$ :

- vitesse  $x_1 \stackrel{\text{def}}{=} v$
- tilt  $x_2 \stackrel{\text{def}}{=} e_z R^T$

# Obtenir une équation diff. qui fait apparaître le tilt

On considère un vecteur  $x$ :

- vitesse  $x_1 \stackrel{\text{def}}{=} v$
- tilt  $x_2 \stackrel{\text{def}}{=} e_z R^T$

Par calcul (en ROCQ) :

$$\begin{cases} \dot{x}_1 = x_1 S(\omega) + y_a - g_0 x_2 & (1) \\ \dot{x}_2 = x_2 S(\omega) & (2) \end{cases}$$

## Obtenir une équation diff. qui fait apparaître le tilt

On considère un vecteur  $x$ :

- vitesse  $x_1 \stackrel{\text{def}}{=} v$
- tilt  $x_2 \stackrel{\text{def}}{=} e_z R^T$

Par calcul (en ROCQ) :

$$\begin{cases} \dot{x}_1 = x_1 S(\omega) + y_a - g_0 x_2 & (1) \\ \dot{x}_2 = x_2 S(\omega) & (2) \end{cases}$$

Par expérience du roboticien :

*Estimateur en deux temps du tilt*

$$\begin{cases} \dot{\hat{x}}_1 = \hat{x}_1 S(\omega) + y_a - g_0 \hat{x}'_2 \\ \dot{\hat{x}}'_2 = -\frac{\alpha_1}{g_0} (v - \hat{x}_1) \\ \dot{\hat{x}}_2 = \hat{x}_2 S(\omega + \gamma \hat{x}'_2 S(\hat{x}_2)) \end{cases} \quad (3)$$

## Dernière étape : Dynamique des erreurs d'estimation

- Erreurs d'estimation:

$$z_{p1} \stackrel{\text{def}}{=} (x_2 - \hat{x}'_2) R^T \quad z_2 \stackrel{\text{def}}{=} (x_2 - \hat{x}_2) R^T$$

- Finalement l'équation différentielle d'intérêt:

$$\begin{cases} \dot{z}_{p1} = -\alpha_1 z_{p1} \\ \dot{z}_2 = \gamma(z_2 - z_{p1}) S^2(e_z - z_2) \end{cases} \quad (4)$$

# Plan

- 1 Introduction
- 2 L'estimation et sa formalisation
- 3 Formalisation de la stabilité**
- 4 Application
- 5 Conclusion

# Plan de la formalisation

- 1 Définir l'équation différentielle `tilt_eqn`
- 2 Caractériser l'espace d'états,  
i.e., l'ensemble des points que peut atteindre une solution
- 3 Prouver la convergence des solutions vers deux points d'équilibre  
(0, 0) et (0,  $2e_z$ )

# Plan de la formalisation

- 1 Définir l'équation différentielle `tilt_eqn`
- 2 Caractériser l'espace d'états,  
i.e., l'ensemble des points que peut atteindre une solution
- 3 Prouver la convergence des solutions vers deux points d'équilibre  
(0, 0) et  $(0, 2e_z)$

On formalise (1), (2), et une partie de (3) : la non-divergence.

# Plan de la formalisation

- 1 Définir l'équation différentielle `tilt_eqn`
- 2 Caractériser l'espace d'états,  
i.e., l'ensemble des points que peut atteindre une solution
- 3 Prouver la convergence des solutions vers deux points d'équilibre  
(0, 0) et (0,  $2e_z$ )

On formalise (1), (2), et une partie de (3) : la non-divergence.

- On a besoin de définir des outils mathématiques...

# Formalisation des équations différentielles

## Solution d'une équation différentielle

Une fonction  $f$  est solution d'une équation différentielle définie par  $\phi$  si :

- à chaque instant  $t$  la dérivée par rapport au temps  $D_1 f \stackrel{\text{def}}{=} \frac{df}{dt}$  est égale à  $\phi(f(t))$ , et
- $f(0)$  satisfait une certaine condition initiale

En Rocq :

```

Let T := 'rV[K]_n. (* le type des vecteurs de longueur n *)
Variable phi : (K -> T) -> K -> T. (* l'équa. diff. *)
Definition is_sol (Init : set T) (x : K -> T) :=
  [/\ x 0 \in Init, (forall t, derivable x t 1) &
    (forall t, 'D_1 x t = phi x t)].
  
```

# Formalisation des équations différentielles

[Cohen and Rouhling, 2017]

- Champ des solutions : prend une condition initiale  $x_0 : 'rV[K]_n$  et un temps  $t : K$

**Variable** `sol` : `'rV[K]_n -> K -> 'rV[K]_n.`

# Formalisation des équations différentielles

[Cohen and Rouhling, 2017]

- Champ des solutions : prend une condition initiale  $x_0 : 'rV[K]_n$  et un temps  $t : K$

**Variable** `sol` : `'rV[K]_n -> K -> 'rV[K]_n`.

- Condition initiale

**Hypothesis** `sol0` : `forall x0 : 'rV[K]_n, sol x0 (0 : K) = x0`.

# Formalisation des équations différentielles

[Cohen and Rouhling, 2017]

- Champ des solutions : prend une condition initiale

$x_0 : 'rV[K]_n$  et un temps  $t : K$

**Variable**  $sol : 'rV[K]_n \rightarrow K \rightarrow 'rV[K]_n$ .

- Condition initiale

**Hypothesis**  $sol_0 : \text{forall } x_0 : 'rV[K]_n, sol\ x_0\ (0 : K) = x_0$ .

- Existence et unicité

**Hypothesis**  $solP : \text{forall } f : K \rightarrow 'rV[K]_n, f\ 0 \ \backslash \text{in } \text{Init} \rightarrow$   
 $is\_sol\ \phi\ \text{Init}\ f \leftrightarrow sol\ (f\ 0) = f$ .

$\Rightarrow$  on travaille sous ces hypothèses

# Stabilité d'un équilibre

## Point d'équilibre

Un *point d'équilibre* est un point  $x$  tel que la fonction constante qui retourne  $x$  est solution de l'équation différentielle  $\phi$  avec un ensemble de positions initiales  $\text{Init}$

```
Definition is_equilibrium_point (x : 'rV[K]_n) :=  
  is_sol phi Init (cst x).
```

## Stabilité au sens de Lyapunov d'un point

On peut toujours circonscrire une trajectoire à un voisinage de  $x$  en choisissant un voisinage de départ approprié

```
Definition is_stable_at (x : 'rV[K]_n) (z : K -> 'rV[K]_n) :=  
  forall eps, eps > 0 -> exists2 d, d > 0 & `| z 0 - x | < d ->  
    forall t, t >= 0 -> `| z t - x | < eps.
```

## Fonction (candidate) de Lyapunov

Comment prouver la stabilité ?

Il s'agit de trouver une fonction dite de Lyapunov.

## Fonction (candidate) de Lyapunov

Comment prouver la stabilité ?

Il s'agit de trouver une fonction dite de Lyapunov.

### Fonction candidate de Lyapunov

Fonction strictement positive en dehors du point d'équilibre et égale à zéro au point d'équilibre.

**Definition** `is_Lyapunov_candidate`

```
{K : realType} (T : normedModType K) (V : T -> K)
(D : set T) (x : T) : Prop :=
x \in D /\ V x = 0 /\ forall z, z \in D -> z != x -> V z > 0.
```

Devient *fonction de Lyapunov* si

```
forall x, x \in Init -> forall t, t >= 0 -> 'D~(sol, x) V t <= 0
```

'D~(sol,x) V

: dérivée de V le long de sol

# Le théorème de stabilité de Lyapunov

## Théorème de stabilité de Lyapunov

**Hypothesis**  $V_{diff}$  :

`forall t : 'rV[K]_n.+1, differentiable V t.`

**Hypothesis**  $V'_{le0}$  : `forall x, x \in Init ->`

`forall t, t >= 0 -> 'D~(sol, x) V t <= 0.`

**Theorem** `Lyapunov_stability (x : 'rV[K]_n.+1 := 0)`

`is_Lyapunov_candidate V Init x ->`

`is_equilibrium_point phi Init x ->`

`is_stable_at x (sol x).`

# Plan

- 1 Introduction
- 2 L'estimation et sa formalisation
- 3 Formalisation de la stabilité
- 4 Application**
- 5 Conclusion

# Application du théorème de Lyapunov au problème

On prouve alors que `point1`, i.e.,  $((0, 0, 0), (0, 0, 0))$  est stable :

`Lemma equilibrium_zero_stable :`

```
open Init -> 0 \in Init -> Init `<=` state_space_tilt ->  
is_stable_at point1 (sol 0).
```

Il faut donc :

# Application du théorème de Lyapunov au problème

On prouve alors que `point1`, i.e.,  $((0, 0, 0), (0, 0, 0))$  est stable :

**Lemma** `equilibrium_zero_stable` :

```
open Init -> 0 \in Init -> Init `<=` state_space_tilt ->
is_stable_at point1 (sol 0).
```

Il faut donc :

- Prouver que `point1` est `is_equilibrium_point`, via un calcul symbolique

# Application du théorème de Lyapunov au problème

On prouve alors que `point1`, i.e.,  $((0, 0, 0), (0, 0, 0))$  est stable :

**Lemma** `equilibrium_zero_stable` :

```
open Init -> 0 \in Init -> Init `<=` state_space_tilt ->
is_stable_at point1 (sol 0).
```

Il faut donc :

- Prouver que `point1` est `is_equilibrium_point`, via un calcul symbolique
- Prouver `is_Lyapunov_candidate...`

# La fonction de Lyapunov

- Benallegue et al. proposent la fonction de Lyapunov suivante:

$$V_1 \stackrel{\text{def}}{=} \frac{\|z_{p1}\|^2}{2\alpha_1} + \frac{\|z_2\|^2}{2\gamma}$$

## La fonction de Lyapunov

- Benallegue et al. proposent la fonction de Lyapunov suivante:  

$$V_1 \stackrel{\text{def}}{=} \frac{\|z_{p1}\|^2}{2\alpha_1} + \frac{\|z_2\|^2}{2\gamma}$$
- Prouvons qu'elle est une fonction de Lyapunov en utilisant les propriétés des formes quadratiques :

$$\dot{V}_1 \leq - \underbrace{\left[ \|z_{p1}\| \quad \|z_2 \mathcal{S}(e_z)\| \right]}_{u_1} \underbrace{\begin{bmatrix} 1 & -1/2 \\ -1/2 & 1 \end{bmatrix}}_{u_2} \begin{bmatrix} \|z_{p1}\| \\ \|z_2 \mathcal{S}(e_z)\| \end{bmatrix} \leq 0. \quad (5)$$

## La fonction de Lyapunov

- Benallegue et al. proposent la fonction de Lyapunov suivante:  

$$V_1 \stackrel{\text{def}}{=} \frac{\|z_{p1}\|^2}{2\alpha_1} + \frac{\|z_2\|^2}{2\gamma}$$
- Prouvons qu'elle est une fonction de Lyapunov en utilisant les propriétés des formes quadratiques :

$$\dot{V}_1 \leq - \underbrace{\left[ \begin{array}{cc} \|z_{p1}\| & \|z_2 S(e_z)\| \end{array} \right]}_{u_1} \underbrace{\left[ \begin{array}{cc} 1 & -1/2 \\ -1/2 & 1 \end{array} \right]}_{u_2} \left[ \begin{array}{c} \|z_{p1}\| \\ \|z_2 S(e_z)\| \end{array} \right] \leq 0. \quad (5)$$

**Lemma** `u2_quadratic_form_gt0` (`v : 'rV_2`) :  
`v != 0 -> 0 < (v *m u2 *m v^T) 0 0.`

## La fonction de Lyapunov

- Benallegue et al. proposent la fonction de Lyapunov suivante:  

$$V_1 \stackrel{\text{def}}{=} \frac{\|z_{p1}\|^2}{2\alpha_1} + \frac{\|z_2\|^2}{2\gamma}$$
- Prouvons qu'elle est une fonction de Lyapunov en utilisant les propriétés des formes quadratiques :

$$\dot{V}_1 \leq - \underbrace{\left[ \|z_{p1}\| \quad \|z_2 S(e_z)\| \right]}_{u_1} \underbrace{\begin{bmatrix} 1 & -1/2 \\ -1/2 & 1 \end{bmatrix}}_{u_2} \begin{bmatrix} \|z_{p1}\| \\ \|z_2 S(e_z)\| \end{bmatrix} \leq 0. \quad (5)$$

**Lemma** `u2_quadratic_form_gt0` (`v : 'rV_2`) :  
`v != 0 -> 0 < (v *m u2 *m v^T) 0 0.`

- Qed.

# Plan

- 1 Introduction
- 2 L'estimation et sa formalisation
- 3 Formalisation de la stabilité
- 4 Application
- 5 Conclusion**

# Depuis la soumission

Collaboration avec Holger Thies (Kyoto U.) et Yoshihiro Ishiguro (Nagoya U./AIST)

## 1 Nous avons formalisé le théorème de Cauchy-Lipschitz:

```
Theorem cauchy_lipschitz_local : exists f delta,
  delta > 0 /\
  f a = u0 /\
  {in `]a, a + delta[, forall x, derivable f x 1 /\ f``() x = phi x (f x)} /\
  {in `[a, a + delta], forall t, closed_ball u0 r%:num (f t)} /\
  {within `[a, a + delta], continuous f}.
```

- Conséquence du théorème du point fixe de Banach et du théorème fondamental de l'analyse (présenté aux JFLA 2024!)
- Utilisation des quotients de MATHCOMP [Cohen, 2013]

## 2 Nous avons éliminé les axiomes de notre expérience

- en utilisant Cauchy-Lipschitz
- en spécialisant un lemme technique à propos des matrices définies positives (qu'il faudrait cependant généraliser...)

# Contributions quantifiées

À titre indicatif

PR: <https://github.com/affeldt-aist/coq-robot/pull/43>

(aussi PRs mineures to MATHCOMP-ANALYSIS déjà mergées: dérivation de la racine carrée, etc.)

<b>fichier</b>	<b>contenu</b>	<b>l.o.c.</b>
<code>derive_matrix.v</code>	dérivation de matrices	713
<code>tilt_analysis.v</code>	extensions de MATHCOMP-ANALYSIS	192
<code>tilt_robot.v</code>	extensions de ROBOTROCQ :	347
	dérivée de la norme euclidienne, etc.	
<code>tilt.v</code>	formalisation de Benallegue et al.:	1918
	théorème de Lyapunov, théorème des valeurs extrêmes pour les vecteurs, etc.	

(Ajout récent: Formalisation de Cauchy-Lipschitz, 3143 l.o.c.)

## Travaux reliés

- Principe de LaSalle en ROCQ [Cohen and Rouhling, 2017], application au problème 2D du pendule inversé [Rouhling, 2018].
- Formalisation d'un équivalent du théorème de stabilité de Lyapunov en ROCQ exprimé au moyen de modalités [Ricketts et al., 2016]. Preuve de la stabilité d'un régulateur P [Chan et al., 2016].
- Stabilité de Lyapunov en Minlog pour des systèmes polynomiaux en temps discret [Devadze et al., 2025].
- D'autres applications de ROCQ en robotique: vérification de protocoles pour robots autogérés [Courtieu et al., 2022], planification de mouvement en 2D [Bertot and Portet, 2025].
- Une formalisation des équations différentielles en temps discret en ROCQ [Thies, 2026]

- [Affeldt and Cohen, 2017] Affeldt, R. and Cohen, C. (2017).  
Formal foundations of 3D geometry to model robot manipulators.  
In *6th ACM SIGPLAN Conference on Certified Programs and Proofs (CPP 2017), Paris, France, January 16–17, 2017*, pages 30–42. ACM.
- [Affeldt et al., 2020] Affeldt, R., Cohen, C., Kerjean, M., Mahboubi, A., Rouhling, D., and Sakaguchi, K. (2020).  
Competing inheritance paths in dependent type theory: A case study in functional analysis.  
In *10th International Joint Conference on Automated Reasoning (IJCAR 2020), Paris, France, July 1–4, 2020*, volume 12167 of *Lecture Notes in Computer Science*, pages 3–20. Springer. Part II.
- [Affeldt et al., 2018] Affeldt, R., Cohen, C., and Rouhling, D. (2018).  
Formalization techniques for asymptotic reasoning in classical analysis.  
*J. Formaliz. Reason.*, 11(1):43–76.
- [Affeldt and Stone, 2024] Affeldt, R. and Stone, Z. (2024).  
A comprehensive overview of the Lebesgue differentiation theorem in Coq.  
In *15th International Conference on Interactive Theorem Proving (ITP 2024), September 9–14, 2024, Tbilisi, Georgia*, volume 309 of *LIPICs*, pages 5:1–5:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- [Benallegue et al., 2023] Benallegue, M., Benallegue, A., Cisneros, R., and Chitour, Y. (2023).  
Velocity-aided IMU-based tilt and attitude estimation.  
*IEEE Trans. Autom. Control.*, 68(10):5823–5836.  
Disponible à l'adresse  
[https://hal.science/hal-04271257v1/file/benallegue2019tac\\_October\\_2022.pdf](https://hal.science/hal-04271257v1/file/benallegue2019tac_October_2022.pdf).

- [Bertot and Portet, 2025] Bertot, Y. and Portet, T. (2025).  
Formally verifying a vertical cell decomposition algorithm.  
In *16th International Conference on Interactive Theorem Proving (ITP 2025) September 28–October 1, 2025, Reykjavik, Iceland*, volume 352 of *LIPICs*, pages 24:1–24:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- [Chan et al., 2016] Chan, M., Ricketts, D., Lerner, S., and Malecha, G. (2016).  
Formal verification of stability properties of cyber-physical systems.  
In *The Second International Workshop on Coq for PL (CoqPL 2016), St. Petersburg, Florida, USA, January 23, 2016*.  
2 pages.
- [Cohen, 2013] Cohen, C. (2013).  
Pragmatic quotient types in coq.  
In *Interactive Theorem Proving - 4th International Conference, ITP 2013, Rennes, France, July 22-26, 2013. Proceedings*, volume 7998 of *Lecture Notes in Computer Science*, pages 213–228. Springer.
- [Cohen and Rouhling, 2017] Cohen, C. and Rouhling, D. (2017).  
A formal proof in Coq of LaSalle's invariance principle.  
In *8th International Conference on Interactive Theorem Proving (ITP 2017), Brasília, Brazil, September 26–29, 2017*, volume 10499 of *Lecture Notes in Computer Science*, pages 148–163. Springer.
- [Courtieu et al., 2022] Courtieu, P., Rieg, L., Tixeuil, S., and Urbain, X. (2022).  
Swarms of Mobile Robots: Towards Versatility with Safety.  
*Leibniz Transactions on Embedded Systems*, 8(2):02:1–02:36.

- [Devadze et al., 2025] Devadze, G., Magron, V., and Streif, S. (2025).  
Computer-assisted proofs for Lyapunov stability via sums of squares certificates and constructive analysis.  
*J. Autom. Reason.*, 69(1):2.
- [Ricketts et al., 2016] Ricketts, D., Malecha, G., and Lerner, S. (2016).  
Modular deductive verification of sampled-data systems.  
In *2016 International Conference on Embedded Software (EMSOFT 2016), Pittsburgh, Pennsylvania, USA, October 1–7, 2016*, pages 17:1–17:10. ACM.
- [Rouhling, 2018] Rouhling, D. (2018).  
A formal proof in Coq of a control function for the inverted pendulum.  
In *7th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2018), Los Angeles, CA, USA, January 8–9, 2018*, pages 28–41. ACM.
- [Thies, 2026] Thies, H. (2026).  
Computing solutions for systems of multivariate ordinary differential equations in rocq.  
CPP '26, page 29–44. Association for Computing Machinery.