

Quantification en commande

Florent Koudohode

18 janvier 2026

Table des matières

1	Motivation et idée générale	2
1.1	Définition formelle d'un quantificateur	2
1.1.1	Seuils de saturation et précision	4
1.1.2	Quantification avec variable de zoom	5
1.2	Exemples de quantification	5
1.2.1	Exemple 1 : quantificateur uniforme scalaire	5
1.2.2	Exemple 2 : quantification logarithmique	6
1.2.3	Exemple 3 : quantificateur à zone morte (deadzone) + niveaux	6
1.2.4	Exemple 4 (bonus) : PWM / actionneur tout-ou-rien (quantification d'entrée)	7
2	Modèle linéaire : état quantifié et retour d'état	8
2.1	Preliminaire : Lyapunov pour $A + BK$	8
2.2	Commande quantifiée par « certitude équivalente »	8
2.3	Analyse : stabilité pratique pour μ fixé	8
2.4	Régions invariantes ellipsoïdales	9
3	Stabilisation asymptotique globale par stratégie hybride (zooming)	9
3.1	Stratégie en deux phases	9
3.2	Exemple numérique : retour d'état quantifié avec zooming (figures)	10
3.3	Comment lire les résultats	13
4	Ouverture : quantification d'entrée, sortie, observateurs	13
5	Résumé (à retenir)	13
6	Rappels : quantification, erreur, saturation, zoom	14
6.1	Définition (générale) et interprétation géométrique	14
6.2	Hypothèses standards (plage / saturation détectable)	14
6.3	Exemple canonique en dimension 1 : quantifieur uniforme (mid-tread, saturant)	14
6.4	Variable de zoom : famille q_μ	15
7	Le système EDP de transport et le contrôle nominal backstepping	15
7.1	Modèle (transport avec couplage de bord)	15
7.2	Transformation backstepping et système cible	15
7.3	Cas explicite (simulation) : $\bar{g} \equiv 0$, $g(x) \equiv g$ constant	15
8	(A) Quantification sur l'état (mesures) : loi hybride + preuves	16
8.1	Loi de commande avec état quantifié	16
8.2	Principe de preuve : deux lemmes (capture + décroissance sur un intervalle)	16
8.3	Théorème (stabilité globale asymptotique)	17

9 (B) Quantification sur l'entrée : $U(t) = q_\mu(U_{\text{nom}}(t))$	17
9.1 Loi hybride (article, Section 4)	17
9.2 Structure de preuve (analogue au cas quantification d'état)	17
10 Quantification de sortie : recette (observer/backstepping) + parallèles	18
11 Simulation : reproduire les figures (norme, cartes, $U(t)$, $\mu(t)$) en Python	18
11.1 Ce qu'on vise (proche des Figures 3–5)	18
11.2 Choix numériques	19
11.3 Code Python (un fichier) : <code>transport_quantized_control.py</code>	19

1 Motivation et idée générale

En théorie de contrôle, on conçoit souvent des lois de retour sous l'hypothèse implicite que les signaux (mesures et commandes) sont réels, continus et transmis sans perte. En pratique, un *ordinateur numérique*, un *convertisseur analogique-numérique* (CAN/ADC), un *réseau de communication* ou un *actionneur discret* impose une **quantification** : une grandeur réelle est représentée par une valeur issue d'un ensemble *discret* de niveaux. Cette discrétisation est structurelle en commande numérique et en contrôle sous contraintes d'information (voir, par exemple, [1, 2, 3, 4]).

Boucle de commande numérique

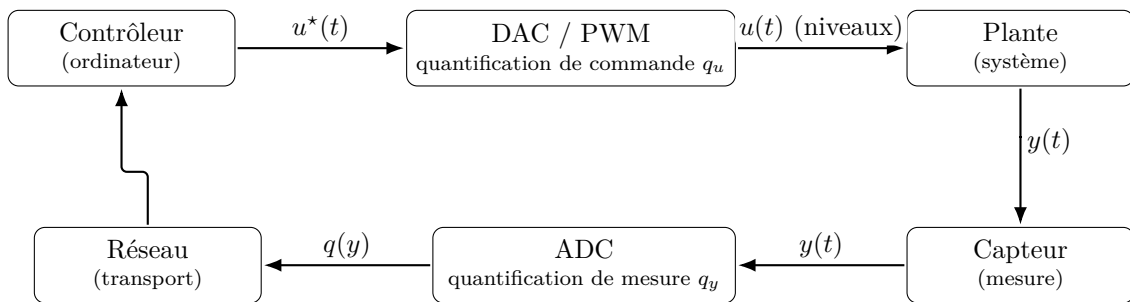


FIGURE 1 – **Où intervient la quantification dans une boucle numérique.** La mesure $y(t)$ est convertie en une valeur numérique (ADC) et devient $q(y)$; elle peut transiter sur un réseau ; puis la commande calculée $u^*(t)$ est convertie en une action réellement appliquée $u(t)$ (DAC ou PWM), souvent à niveaux.

Signes :

ADC (Analog-to-Digital Converter) : convertisseur analogique-numérique. Il transforme un signal continu en un nombre (souvent codé sur N bits).

DAC (Digital-to-Analog Converter) : convertisseur numérique-analogique. Il transforme une valeur numérique (discrète) en tension/courant analogique.

PWM (Pulse-Width Modulation) : modulation de largeur d'impulsion. On applique une commande “tout ou rien” très rapide dont la valeur moyenne sur une période approxime une commande continue.

1.1 Définition formelle d'un quantificateur

Définition 1 (Quantificateur (partition en régions)). Soit $z \in \mathbb{R}^\ell$. Un **quantificateur** est une application

$$q : \mathbb{R}^\ell \rightarrow \mathcal{Q}, \quad \mathcal{Q} \subset \mathbb{R}^\ell \text{ fini (ou au moins dénombrable),}$$

telle que q soit *par morceaux constante*. Autrement dit, l'espace $\mathbb{R}^\ell$ est découpé en **régions de quantification**

$$\mathbb{R}^\ell = \bigcup_{i \in I} \mathcal{R}_i, \quad \mathcal{R}_i \cap \mathcal{R}_j = \emptyset \ (i \neq j),$$

et $q(z)$ prend une **valeur de reconstruction** constante sur chaque région : $q(z) = q_i$ pour tout $z \in \mathcal{R}_i$.

Remarque 1. On peut voir q comme une *mesure numérique* : au lieu de transmettre z exactement, on transmet l'indice de la région (la “cellule”) dans laquelle se trouve z , puis on reconstruit une valeur représentative q_i . C'est précisément l'interprétation visuelle donnée par une partition en régions (Figure 2).

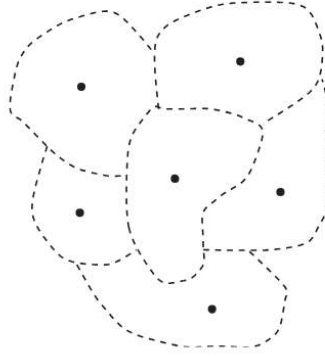


FIGURE 2 – **Régions de quantification** : exemple de partition de l'espace en cellules où q est constant, voir [4].

1.1.1 Seuils de saturation et précision

On caractérise un quantificateur par deux paramètres opérationnels :

- $M > 0$: **seuil de saturation** (ou **borne de saturation** ou **plage**). Tant que $\|z\| \leq M$, le quantificateur ne sature pas ; au-delà, l'information disponible devient insuffisante (écrêtage).
- $\Delta > 0$: **précision** (ou **erreur de quantification**) lorsque le quantificateur ne sature pas.

Pourquoi ces deux paramètres sont-ils essentiels ?

- Δ mesure l'*incertitude injectée* par la quantification en régime non saturé : on peut modéliser $q(z) = z + e(z)$ avec $\|e(z)\| \leq \Delta$.
- M fixe les *seuils de saturation* : si le signal franchit ces seuils, l'erreur n'est plus contrôlable par Δ et certaines lois de commande peuvent perdre leur efficacité.

Voici, quelques hypothèses (classiques et très utilisées pour les contrôles quantifiés) :

(P1) (Erreur bornée hors saturation) : si $\|z\| \leq M$, alors

$$\|q(z) - z\| \leq \Delta.$$

(P2) (Saturation détectable) : si $\|z\| > M$, alors

$$\|q(z)\| > M - \Delta.$$

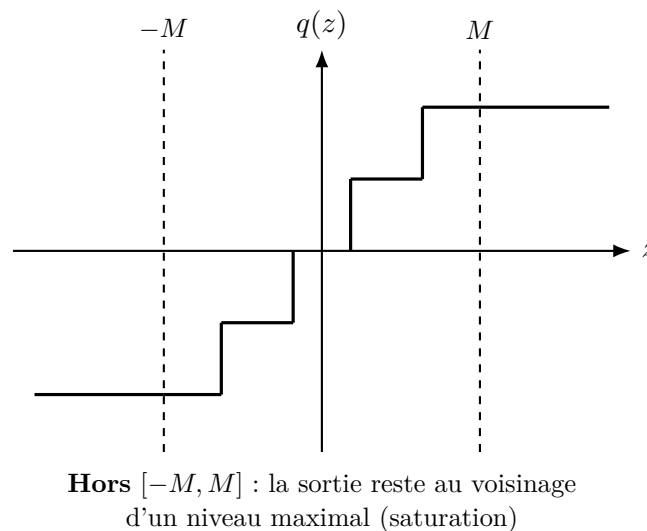


FIGURE 3 – **Saturation.** Lorsque $|z|$ dépasse M , le quantificateur ne peut plus représenter correctement les grandes valeurs : la sortie sature et reste proche d'un niveau maximal. *Source* : inspirée des discussions sur quantification et information limitée dans [4, 3].

Remarque 2 (Lien avec les “régions de quantification”). La Fig. 2 illustre l’idée géométrique : l’espace est découpé en régions (cellules) et q renvoie un niveau constant par région. L’hypothèse (P1) signifie qu’à l’intérieur d’un ensemble borné (délimité par les *seuils de saturation* $\|z\| \leq M$), la cellule choisie garantit une reconstruction à erreur au plus Δ . L’hypothèse (P2) encode le comportement au-delà : le quantificateur *sature* et la sortie “colle” près des seuils.

1.1.2 Quantification avec variable de zoom

On définit une famille de quantificateurs indexée par $\mu > 0$:

$$q_\mu(z) := \mu q\left(\frac{z}{\mu}\right).$$

Alors :

- les **seuils de saturation** deviennent $\pm M\mu$;
- la **précision** (l’erreur de quantification) devient $\Delta\mu$.

Intuitivement

- **Zoom-out** : augmenter μ pour repousser les seuils de saturation $\pm M\mu$ lorsque l’état est loin (éviter l’écrêtage).
- **Zoom-in** : diminuer μ par paliers une fois le système “capturé” dans une zone sûre ; la borne d’erreur $\Delta\mu$ décroît, ce qui permet une convergence plus fine.

Une intuition utile est l’analogie avec l’imagerie numérique : une scène réelle est continue, mais une caméra la représente par un nombre fini de pixels et de niveaux (résolution finie). Cette idée reflète exactement le rôle d’un quantificateur : convertir un signal réel en une représentation à alphabet fini. De plus, la notion de *zoom* illustre l’idée clé des stratégies de *zooming* en commande quantifiée : on peut ajuster un paramètre d’échelle pour éviter la saturation loin de l’origine (*zoom-out*, zoomer), puis améliorer la précision près de l’équilibre (*zoom-in*, dézoomer).

1.2 Exemples de quantification

1.2.1 Exemple 1 : quantificateur uniforme scalaire

Soit $\ell = 1$. Fixons un pas $h > 0$.

$$q(z) = h \cdot \left\lfloor \frac{z}{h} + \frac{1}{2} \right\rfloor, \quad \lfloor x \rfloor = \max\{k \in \mathbb{Z} : k \leq x\},$$

avec saturation éventuelle : si $|z| > M$, on renvoie $\text{sat}_M(z)$.

Interprétation. On remplace z par le *multiple de h le plus proche*. Cela correspond à l’arrondi au pas. En conversion analogique-numérique, h est directement lié au nombre de bits et à la pleine échelle. Une analyse classique donne une erreur bornée par $h/2$ hors saturation [2].

- Erreur typique : $|q(z) - z| \leq \frac{h}{2}$ (hors saturation), donc $\Delta = \frac{h}{2}$.



FIGURE 4 – **Analogie “caméra/zoom” pour la quantification et le zooming.** La représentation numérique d’une scène repose sur une résolution finie (alphabet fini), et le zoom illustre une mise à l’échelle qui modifie l’étendue effective et la précision. Cette intuition correspond au quantificateur à variable de zoom $q_\mu(z) = \mu q(z/\mu)$. *Source* : illustration pédagogique fournie par l’auteur ; l’analogie “caméra/zoom” est également discutée dans la littérature sur le zooming en commande quantifiée [4, 3].

1.2.2 Exemple 2 : quantification logarithmique

Pour $\ell = 1$ et $z \neq 0$:

$$q(z) = \text{sign}(z) \alpha^k, \quad k = \lfloor \log_\alpha(|z|) \rfloor, \quad \alpha > 1.$$

L’erreur est *relative* : typiquement $|q(z) - z| \leq c |z|$ dans certaines zones.

Contrairement au quantificateur uniforme (erreur absolue), ici les niveaux sont espacés *multiplicativement*. Les petites valeurs sont quantifiées très finement, tandis que les grandes valeurs sont codées plus grossièrement, avec un contrôle en *pourcentage*. Cette idée est classique côté quantification, et apparaît aussi dans des schémas modernes (p.ex. quantification log-échelle en apprentissage machine) [2, 5, 6].

1.2.3 Exemple 3 : quantificateur à zone morte (deadzone) + niveaux

Très utilisé en actionnement discret :

$$q(z) = \begin{cases} 0, & |z| \leq d, \\ h \cdot \left\lfloor \frac{z}{h} + \frac{1}{2} \right\rfloor, & |z| > d, \end{cases}$$

avec $d > 0$ (zone morte). Cela modélise un capteur qui ne « voit » rien sous un seuil, ou un actionneur avec frottement / jeu.

La zone morte est une non-linéarité très courante en pratique. Elle introduit une perte d’information près de zéro : tant que $|z| \leq d$, la sortie est exactement nulle. Cela peut ralentir la convergence, voire créer un biais si la commande idéale exige de petites corrections. Des discussions générales sur ce type de non-linéarités (deadzone, saturation) se trouvent classiquement en automatique non linéaire [7].

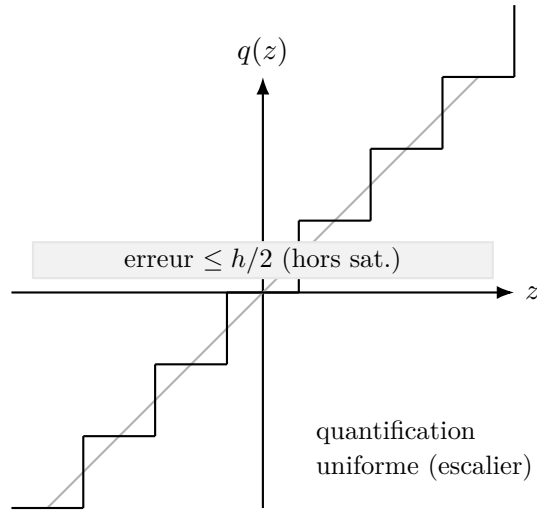


FIGURE 5 – **Quantificateur uniforme (escalier)** : par morceaux constant, erreur bornée hors saturation. *Source* : figure originale ; bornes d’erreur et théorie générale : [2].

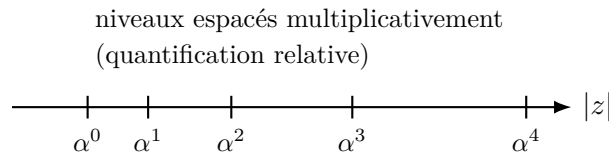


FIGURE 6 – **Quantification logarithmique (schéma 1D)** : niveaux espacés par un facteur $\alpha > 1$, typiquement associée à une erreur relative. *Source* : figure originale ; voir [2, 5, 6].

1.2.4 Exemple 4 (bonus) : PWM / actionneur tout-ou-rien (quantification d’entrée)

Un actionneur ne prend que quelques niveaux : $u \in \{-U_{\max}, 0, U_{\max}\}$, ou $\{0, U_{\max}\}$ (valves, relais, hacheurs). On peut voir cela comme une quantification de l’entrée :

$$u = q(u^*),$$

où u^* est la commande « idéale ».

2 Modèle linéaire : état quantifié et retour d'état

On étudie en détail le cas linéaire continu :

$$\dot{x} = Ax + Bu, \quad x \in \mathbb{R}^n, u \in \mathbb{R}^m. \quad (1)$$

Hypothèse standard : (A, B) est stabilisable, donc il existe K tel que $A + BK$ soit Hurwitz.

2.1 Préliminaire : Lyapunov pour $A + BK$

Si $A + BK$ est Hurwitz, alors pour toute matrice symétrique définie positive $Q = Q^\top \succ 0$, il existe une unique $P = P^\top \succ 0$ telle que

$$(A + BK)^\top P + P(A + BK) = -Q. \quad (2)$$

On pose le candidat Lyapunov :

$$V(x) = x^\top P x, \quad \lambda_{\min}(P) \|x\|^2 \leq V(x) \leq \lambda_{\max}(P) \|x\|^2.$$

2.2 Commande quantifiée par « certitude équivalente »

Supposons qu'on ne mesure pas x , mais seulement $q_\mu(x)$. On implémente :

$$u = K q_\mu(x). \quad (3)$$

L'erreur de mesure quantifiée vaut

$$e(x) := q_\mu(x) - x.$$

Le système en boucle fermée devient

$$\dot{x} = (A + BK)x + BK e(x). \quad (4)$$

2.3 Analyse : stabilité pratique pour μ fixé

On suppose ici μ constant et que l'on est hors saturation, c.-à-d. $\|x\| \leq M\mu$. Alors $\|e(x)\| \leq \Delta\mu$.

Lemme 1 (Borne sur $\dot{V}$ et région ultime). *Sous (2)–(4), si $\|x\| \leq M\mu$ alors*

$$\dot{V}(x) \leq -\lambda_{\min}(Q) \|x\|^2 + 2 \|PBK\| \|x\| \Delta\mu.$$

En particulier, dès que

$$\|x\| \geq \frac{2 \|PBK\|}{\lambda_{\min}(Q)} \Delta\mu,$$

on a $\dot{V}(x) \leq 0$ (décroissance stricte au-delà de ce rayon).

Démonstration. En dérivant $V = x^\top P x$ le long de (4) :

$$\dot{V} = x^\top ((A + BK)^\top P + P(A + BK))x + 2x^\top PBK e.$$

En utilisant (2) :

$$\dot{V} = -x^\top Q x + 2x^\top PBK e \leq -\lambda_{\min}(Q) \|x\|^2 + 2 \|PBK\| \|x\| \|e\|.$$

Hors saturation, $\|e\| \leq \Delta\mu$, d'où le résultat. $\square$

Remarque 3 (Conséquence). Avec un quantificateur **fixe** (μ constant), on obtient typiquement une **stabilité pratique** : les trajectoires entrent dans un voisinage de l'origine dont la taille est de l'ordre de $\Delta\mu$ et y restent. Pour rendre ce voisinage arbitrairement petit, il faut diminuer μ au cours du temps (zoom-in).

2.4 Régions invariantes ellipsoïdales

On retrouve (et on prouve) l'idée structurante : il existe deux ellipsoïdes emboîtés $R_2(\mu) \subset R_1(\mu)$ tels que (i) R_1 et R_2 soient invariants, (ii) toute trajectoire partant de R_1 entre dans R_2 en temps fini.

On introduit la constante

$$\gamma := \frac{2 \|PBK\|}{\lambda_{\min}(Q)}.$$

Lemme 2 (Invariance et contraction pour μ fixé). *Fixons $\varepsilon > 0$ et supposons que M est assez grand par rapport à Δ pour satisfaire*

$$\lambda_{\min}(P)M > \lambda_{\max}(P) \gamma \Delta (1 + \varepsilon). \quad (5)$$

Définissons les ellipsoïdes

$$R_1(\mu) := \left\{ x : x^\top P x \leq \lambda_{\min}(P) M^2 \mu^2 \right\},$$

$$R_2(\mu) := \left\{ x : x^\top P x \leq \lambda_{\max}(P) \gamma^2 \Delta^2 (1 + \varepsilon)^2 \mu^2 \right\}.$$

Alors, pour le système (4) (avec μ constant) :

- $R_1(\mu)$ et $R_2(\mu)$ sont **invariants** ;
- toute trajectoire partant de $R_1(\mu)$ atteint $R_2(\mu)$ en **temps fini**.

Preuve commentée. Étape 1 : emboîtement dans des boules. On compare aux boules $\{\|x\| \leq M\mu\}$ (plage) et $\{\|x\| \leq \gamma\Delta(1 + \varepsilon)\mu\}$ (rayon ultime). Grâce aux inégalités de Rayleigh, on a :

$$x^\top P x \leq \lambda_{\min}(P) M^2 \mu^2 \Rightarrow \|x\|^2 \leq M^2 \mu^2,$$

donc $R_1(\mu) \subset \{\|x\| \leq M\mu\}$, ce qui garantit l'absence de saturation dans R_1 . De même, $R_2(\mu)$ contient la boule $\{\|x\| \leq \gamma\Delta(1 + \varepsilon)\mu\}$ via $V(x) \leq \lambda_{\max}(P) \|x\|^2$. La condition (5) impose que $R_2(\mu) \subset R_1(\mu)$.

Étape 2 : décroissance de V entre les deux frontières. Hors saturation, Lemme 1 donne :

$$\dot{V} \leq -\lambda_{\min}(Q) \|x\| (\|x\| - \gamma\Delta\mu).$$

Donc, sur l'anneau $\gamma\Delta(1 + \varepsilon)\mu \leq \|x\| \leq M\mu$, on a $\dot{V} \leq -c_\varepsilon \mu^2$ pour une constante $c_\varepsilon > 0$. Ainsi V décroît strictement jusqu'à entrer dans $R_2(\mu)$ en temps fini.

Étape 3 : invariance. Une fois dans $R_2(\mu)$, si la trajectoire tentait d'en sortir, il faudrait traverser une zone où $\dot{V} < 0$, contradiction (argument standard par barrières de Lyapunov). Même logique pour $R_1(\mu)$. $\square$

3 Stabilisation asymptotique globale par stratégie hybride (zooming)

Le point clé : si l'on peut **réduire** μ au cours du temps, alors $R_2(\mu)$ rétrécit et on force $x(t) \rightarrow 0$.

3.1 Stratégie en deux phases

Phase 1 : zoom-out (en boucle ouverte). On met $u = 0$ et on augmente μ par paliers jusqu'à ce que le quantificateur ne sature plus et que l'état soit « mesurable » avec précision relative suffisante.

Phase 2 : zoom-in (en boucle fermée). On applique $u = Kq_\mu(x)$ et on diminue μ par un facteur $\rho \in (0, 1)$ toutes les T secondes (dwell time), de façon à recaler $R_2(\mu)$ sur $R_1(\rho\mu)$.

Théorème 1 (Stabilisation globale par quantification hybride). *Supposons (1) stabilisable et que M/Δ est suffisamment grand pour satisfaire une condition de type (5) (et un ajustement de paramètres permettant $\rho < 1$). Alors il existe une loi hybride (mise à jour discrète de μ) telle que la boucle fermée soit **globalement asymptotiquement stable**.*

Schéma de preuve (très pédagogique). **(i) Zoom-out.** Avec $u = 0$, $x(t) = e^{At}x(0)$ peut croître. On augmente μ assez vite pour garantir, à un certain instant t_0 , que $\|x(t_0)\| \leq M\mu(t_0)$ (donc pas de saturation). Cela peut être détecté en pratique en vérifiant que la sortie quantifiée ne colle pas à la frontière (selon les propriétés (P1)-(P2)).

(ii) Zoom-in. À partir de t_0 , on fixe $\mu = \mu_0$ sur une durée T suffisante pour garantir, via Lemme 2, que $x(t_0 + T) \in R_2(\mu_0)$. On choisit ensuite $\rho \in (0, 1)$ tel que

$$R_2(\mu_0) \subset R_1(\rho\mu_0).$$

On met $\mu \leftarrow \rho\mu$ et on répète. Ainsi, à des instants $t_k = t_0 + kT$, on a $x(t_k) \in R_1(\mu_k)$ avec $\mu_k = \rho^k \mu_0 \rightarrow 0$. Comme les ensembles rétrécissent vers $\{0\}$, on obtient $x(t) \rightarrow 0$. $\square$

Remarque 4 (Temps de séjour et robustesse). Le fait que μ ne change qu'à des instants discrets rend la stratégie robuste à des délais et à l'échantillonnage. On peut aussi déclencher les mises à jour par événements (entrée dans une région), au lieu de temps fixes.

3.2 Exemple numérique : retour d'état quantifié avec zooming (figures)

Système et loi de commande. On considère le système linéaire

$$\dot{x} = Ax + Bu, \quad A = \begin{pmatrix} 0 & 1 \\ 2 & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

avec un retour d'état stabilisant (conçu hors quantification) et implémenté avec une mesure quantifiée :

$$u(t) = K q_{\mu(t)}(x(t)), \quad K = \begin{pmatrix} -5 & -4 \end{pmatrix}.$$

Le quantificateur est un *uniforme par composante* (pas h) avec saturation en coordonnées scalées (page M), et une variable de zoom $\mu(t)$ mise à jour de manière hybride :

$$q_\mu(x) = \mu q\left(\frac{x}{\mu}\right), \quad \mu \leftarrow \rho\mu \text{ toutes les } T_{\text{dwell}} \text{ secondes en phase de zoom-in.}$$

Paramètres de simulation. Pour la démonstration :

$$M = 6, \quad h = 0.25, \quad \Delta = \frac{h}{2}, \quad \mu_0 = 1, \quad \rho = 0.7, \quad T_{\text{dwell}} = 0.8, \quad \Delta t = 10^{-3}, \quad T_{\text{end}} = 8.$$

La phase *zoom-out* augmente μ tant que la condition de non-saturation n'est pas satisfaite, puis on passe en *zoom-in*.

Lien vers le code Python (Google Colab). Le notebook complet (code exécutable, paramètres, génération des figures) est disponible ici :

https://colab.research.google.com/drive/1I_w4ITAzfgl2Li9CxWecNXx8ixEFHLHG?usp=sharing

Résultats : états, zoom et commande. La Fig. 7 montre la convergence des états, la Fig. 8 l'évolution de $\mu(t)$, et la Fig. 9 la commande quantifiée appliquée.

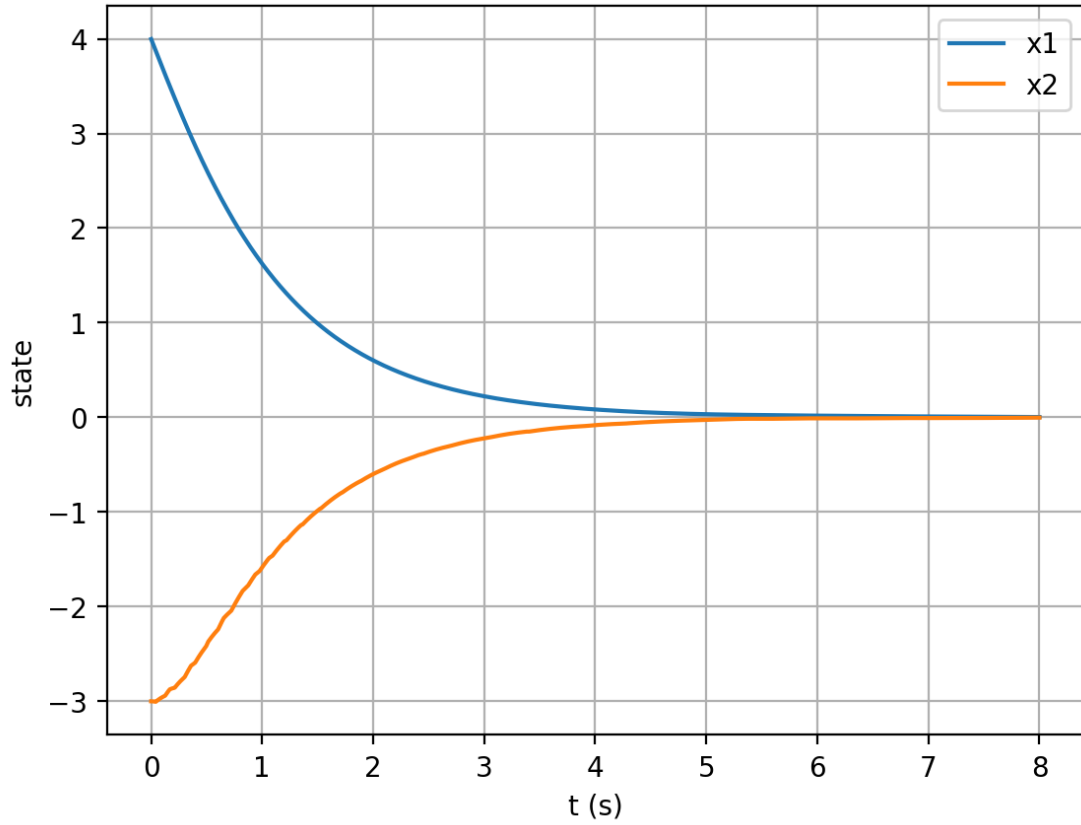


FIGURE 7 – Trajectoires d'état $x_1(t)$ et $x_2(t)$ sous retour quantifié avec zooming.

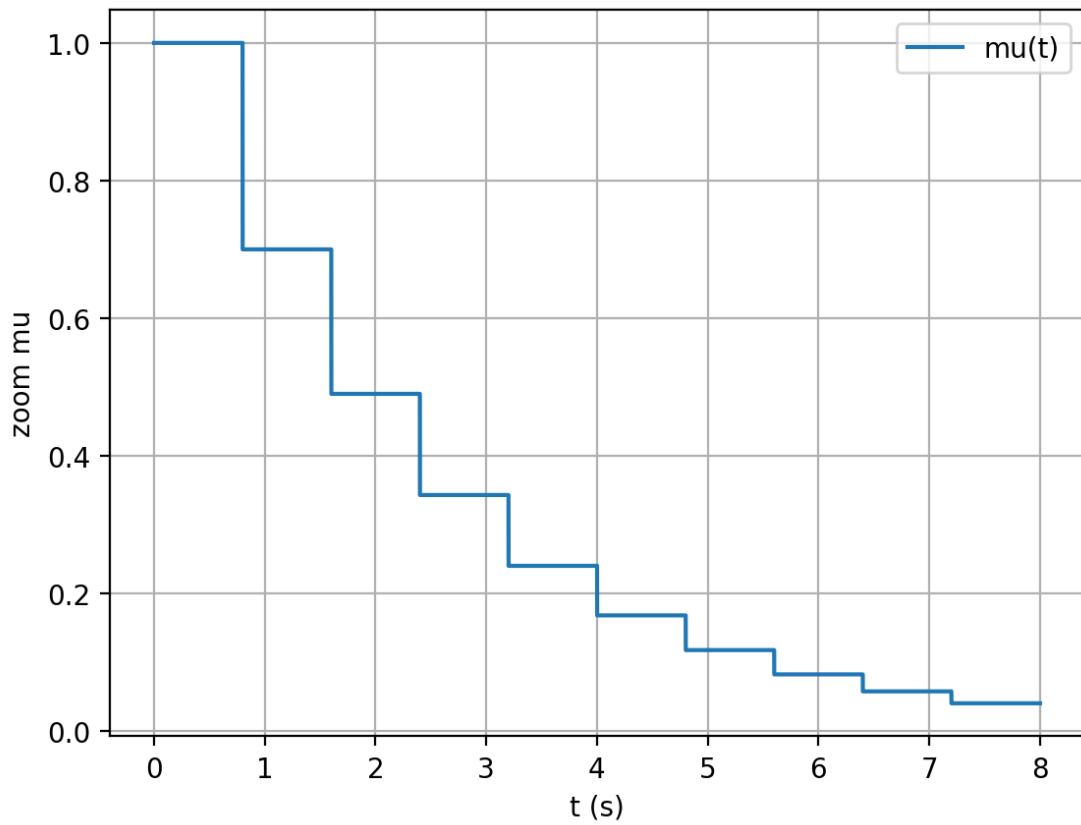


FIGURE 8 – Évolution de la variable de zoom $\mu(t)$ (zoom-out puis zoom-in par paliers).

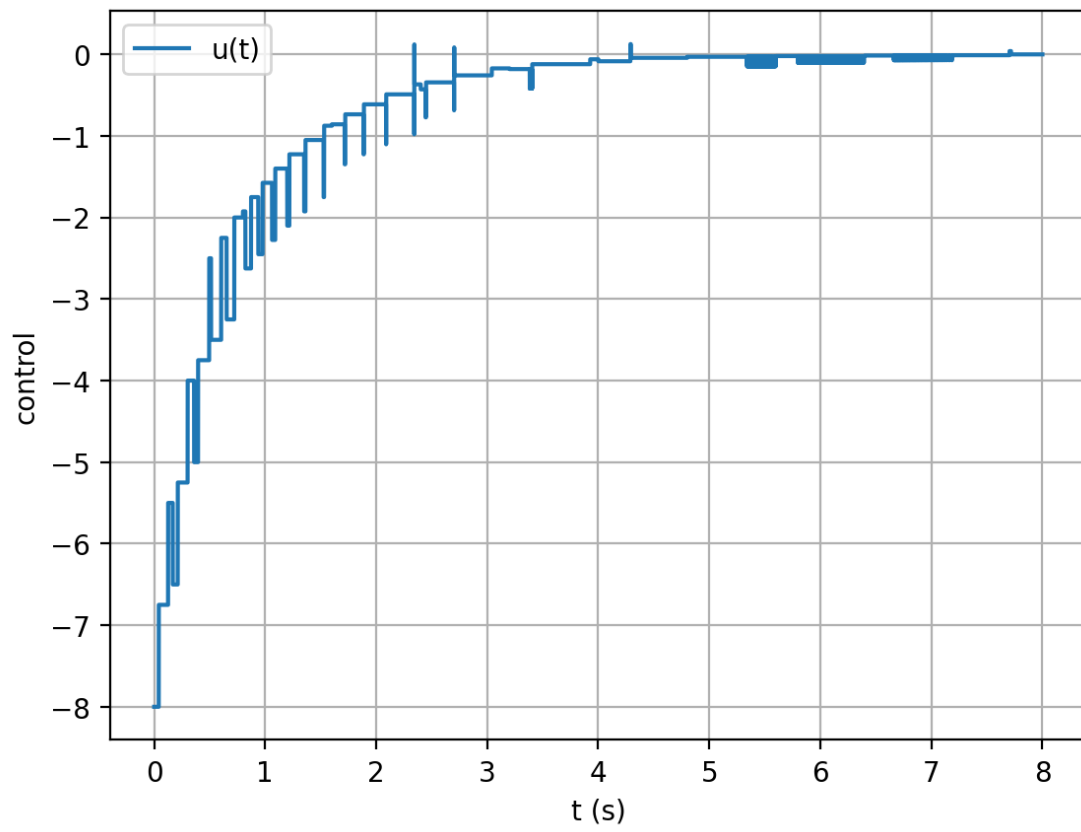


FIGURE 9 – Commande $u(t)$ appliquée (issue de $K q_{\mu(t)}(x(t))$).

3.3 Comment lire les résultats

Vous devriez observer :

- Pendant **zoom-out**, μ augmente jusqu'à éviter la saturation (dans une vraie implémentation, on déduirait cela de la sortie quantifiée).
- En **zoom-in**, μ décroît géométriquement ($\mu \leftarrow \rho\mu$), donc l'erreur $\Delta\mu$ décroît aussi.
- Les états x_1, x_2 convergent vers 0.

Remarque 5 (Choix des paramètres). — Si ρ est trop proche de 1 : convergence plus lente.

- Si T_{dwell} est trop petit : la dynamique n'a pas le temps de « rentrer » dans la région requise avant de réduire μ .
- Si M est trop petit (par rapport à Δ et aux gains) : risque de saturation et la preuve ne s'applique plus.

4 Ouverture : quantification d'entrée, sortie, observateurs

Quantification d'entrée. Si l'entrée est quantifiée ($u = q_\mu(Kx)$), l'analyse est analogue : l'erreur est un *disturbance* en entrée. Souvent, c'est même plus simple conceptuellement car on traite une perturbation additive sur u .

Quantification de sortie et observateur. Si $y = Cx$ est quantifié, on peut combiner un observateur (type Luenberger) avec $q_\mu(y)$, et refaire une analyse Lyapunov sur l'état augmenté $(x, \tilde{x})$ (erreur d'observation). La stratégie de zooming reste applicable.

5 Résumé (à retenir)

1. Un quantificateur impose **erreur** (Δ) et **plage** (M).
2. Avec un quantificateur fixe, on obtient typiquement une **stabilité pratique** : convergence vers une boule d'ordre $\Delta\mu$.
3. Avec un quantificateur à **zoom dynamique** (hybride), on peut obtenir **stabilité asymptotique globale** pour les systèmes linéaires stabilisables.
4. Les preuves s'appuient sur un Lyapunov quadratique et des **régions invariantes emboîtées** qui se rétrécissent lorsque μ décroît.

Cours (continuation) : quantification en commande de l'EDP de transport

Du cas linéaire (dimension finie) au cas hyperbolique (transport) : parallèles, preuves, et simulations

Contexte et fil conducteur. Vous avez déjà présenté un *premier cours* sur la quantification en dimension finie (cas linéaire), avec (i) quantification comme perturbation bornée dans une plage, (ii) saturation hors plage, (iii) variable de zoom μ (zoom-out puis zoom-in) pour concilier *éviter la saturation* et *réduire l'erreur*. Dans ce document, on poursuit *la même logique* pour une EDP de transport, en suivant pas à pas la stratégie du papier de Nikos Bekiaris-Liberis (SCL 2020) *et en explicitant au maximum les idées* (avec parallèles au cas linéaire). :contentReference[oaicite :0]index=0 :contentReference[oaicite :1]index=1

Plan.

1. Rappels quantification : définition, plage / seuil(s) de saturation, erreur, et exemples.

2. EDP de transport considérée, instabilité ouverte, et *contrôle nominal backstepping*.
3. **(A)** Quantification sur l'état (mesures) : loi hybride, preuves (capture + décroissance), conditions de type small-gain.
4. **(B)** Quantification sur l'entrée : $U(t) = \mu q(\bar{U}(t)/\mu)$ avec $\bar{U}(t)$ nominal (backstepping), preuves (adaptées).
5. Quantification de sortie : *cadre et recette* (observer/backstepping) + parallèles (cours + pistes).
6. Simulation numérique : code Python (schémas, figures $\|u(t)\|_\infty$, cartes $u(t, x)$, $U(t)$, $\mu(t)$) proche de l'article.

6 Rappels : quantification, erreur, saturation, zoom

6.1 Définition (générale) et interprétation géométrique

Définition 2 (Quantificateur). Soit $z \in \mathbb{R}^\ell$. Un **quantificateur** est une application à valeurs finies (ou dénombrables)

$$q : \mathbb{R}^\ell \rightarrow \mathcal{Q}, \quad \mathcal{Q} \subset \mathbb{R}^\ell.$$

Typiquement, q est *par morceaux constante* : l'espace $\mathbb{R}^\ell$ est partitionné en *régions de quantification* et q est constant sur chaque région.

Paramètres critiques : seuil(s) de saturation et borne d'erreur. Au lieu de dire *range* (anglais), on utilisera en français :

- $M > 0$: **seuil de saturation** (ou **borne de saturation**) ;
- $\Delta > 0$: **borne d'erreur de quantification** (précision).

L'idée est : *tant que $\|z\| \leq M$ on contrôle l'erreur, sinon on entre en saturation.*

6.2 Hypothèses standards (plage / saturation détectable)

On reprend la formulation standard utilisée dans l'article (et dans votre cours linéaire) : :contentReference[oaicite :2]index=2 :contentReference[oaicite :3]index=3

(P1) (Erreur bornée en régime non saturé) : si $\|z\| \leq M$, alors

$$\|q(z) - z\| \leq \Delta.$$

(P2) (Saturation détectable) : si $\|z\| > M$, alors

$$\|q(z)\| > M - \Delta.$$

Remarque 6. Le point (P2) formalise une idée simple : *si z est trop grand, la sortie “colle” près de la frontière*, donc on peut *détecter* (ou soupçonner) la saturation en observant $q(z)$. C'est exactement la brique qui rend la stratégie *zoom-out / capture* possible.

6.3 Exemple canonique en dimension 1 : quantifieur uniforme (mid-tread, saturant)

Dans le papier SCL, un quantifieur scalaire (uniforme, saturant) est utilisé en simulation, typiquement :

$$q(\xi) = \begin{cases} M, & \xi > M, \\ -M, & \xi < -M, \\ \Delta \left\lfloor \frac{\xi}{\Delta} + \frac{1}{2} \right\rfloor, & -M \leq \xi \leq M. \end{cases}$$

6.4 Variable de zoom : famille q_μ

On définit une famille de quantifieurs par mise à l'échelle :

$$q_\mu(z) = \mu q\left(\frac{z}{\mu}\right), \quad \mu > 0.$$

Alors :

$$\text{seuil de saturation} \sim M\mu, \quad \text{borne d'erreur} \sim \Delta\mu.$$

C'est la clef : *augmenter* μ évite la saturation, *diminuer* μ réduit l'erreur (précision plus fine).

7 Le système EDP de transport et le contrôle nominal backstepping

7.1 Modèle (transport avec couplage de bord)

On considère la classe (scalaire) de l'article SCL :

$$u_t(t, x) = u_x(t, x) + \bar{g}(x) u(t, x) + g(x) u(t, 0), \quad x \in [0, D], \quad t \geq 0, \quad (6)$$

$$u(t, D) = U(t). \quad (7)$$

Cette EDP est *ouverte* potentiellement instable à cause du terme $g(x) u(t, 0)$ (feedback interne provenant de la trace au bord). Le contrôle est appliqué au bord $x = D$.

Remarque 7 (Parallèle au cas linéaire). Dans un système linéaire $\dot{X} = AX + BU$, si A est instable, un retour $U = KX$ stabilise. Ici, l'EDP est l'analogie infini-dimensionnel : on stabilise via une loi au bord $U(t)$ construite pour « casser » le mécanisme d'amplification interne.

7.2 Transformation backstepping et système cible

Le backstepping construit une transformation intégrale inversible :

$$w(t, x) = u(t, x) - \int_0^x k(x, y) u(t, y) dy, \quad (8)$$

$$u(t, x) = w(t, x) + \int_0^x l(x, y) w(t, y) dy, \quad (9)$$

qui transforme (6)–(7) en une EDP « cible » de transport pur :

$$w_t(t, x) = w_x(t, x), \quad x \in [0, D], \quad (10)$$

$$w(t, D) = 0. \quad (11)$$

Cette dynamique est finie-temps stable (les caractéristiques entrent depuis $x = D$, et la condition $w(t, D) = 0$ purge l'état). La loi nominale $U_{\text{nom}}(t)$ est choisie pour imposer (11) via (8)–(9).

7.3 Cas explicite (simulation) : $\bar{g} \equiv 0$, $g(x) \equiv g$ constant

Dans l'exemple numérique de l'article, $D = 1$, $\bar{g} \equiv 0$, $g(x) = g = 1.25$. Dans ce cas, les noyaux sont explicites (et c'est idéal pédagogiquement) :

$$k(x, y) = g e^{g(x-y)}, \quad l(x, y) = g \quad (\text{formes explicites utilisées pour l'exemple constant}).$$

Les constantes d'équivalence de norme (utiles dans les preuves) existent :

$$\|w(t)\|_\infty \leq M_1 \|u(t)\|_\infty, \quad \|u(t)\|_\infty \leq \frac{1}{M_2} \|w(t)\|_\infty,$$

avec des $M_1, M_2 > 0$ dépendant de $(D, g, \bar{g})$ (et explicitables en cas constant). :contentReference[oaicite :8]index=8

Remarque 8. Dans votre cours linéaire, l'analogie est : $|Z| \leq c_1|X|$ et $|X| \leq c_2|Z|$ entre état original et état transformé (changement de coordonnées). Ici, ce sont des estimations ∞ .

8 (A) Quantification sur l'état (mesures) : loi hybride + preuves

8.1 Loi de commande avec état quantifié

On suppose que l'état spatial $u(t, \cdot)$ est mesuré mais *quantifié* (capteur + réseau). On applique une commande backstepping en remplaçant u par $q_\mu(u/\mu)$ dans le retour.

Commande (schéma général). Le papier construit une loi hybride :

$$U(t) = \begin{cases} 0, & 0 \leq t \leq t_1^* \quad (\text{phase zoom-out / open-loop}), \\ \mu(t) \int_0^D k(D, x) q\left(\frac{u(t, x)}{\mu(t)}\right) dx, & t > t_1^*, \end{cases}$$

avec une mise à jour piecewise-constante de μ après capture (zoom-in par paliers). L'objectif est :

- **Phase 1 (zoom-out)** : rendre $M\mu(t)$ assez grand pour que l'état entre dans la zone non saturée.
- **Phase 2 (zoom-in)** : diminuer μ par facteur $\Omega \in (0, 1)$ sur des intervalles de durée T , tout en gardant l'état dans une région où la contraction est garantie.

La construction exacte (choix de t_1^* , τ , T , Ω , et événements de capture) suit Lemma 1, Lemma 2 et Theorem 1 de l'article. :contentReference[oaicite :9]index=9 :contentReference[oaicite :10]index=10

8.2 Principe de preuve : deux lemmes (capture + décroissance sur un intervalle)

On expose la structure standard (exactement le parallèle dimension finie).

Lemme 3 (Capture : existence d'un temps t_1^* où l'état entre dans la plage). *Il existe un temps $t_1^* < \infty$ tel que l'état (transformé) vérifie une condition de type*

$$\frac{\|w(t_1^*)\|_\infty}{\mu(t_1^*)} \leq C_{\text{cap}} M,$$

où C_{cap} dépend des constantes de backstepping $M_1, M_2, \dots$ et des paramètres λ, ν (marges) choisis.

Idée (preuve). Sur $[0, t_1^*]$, on est en boucle ouverte ($U \equiv 0$), donc on peut majorer $\|u(t)\|_\infty$ (et donc $\|w(t)\|_\infty$) par une croissance exponentielle en t . Ensuite, on impose que $\mu(t)$ croisse *plus vite* (ou suffisamment vite) pour forcer le rapport $\|w(t)\|_\infty/\mu(t)$ à entrer sous le seuil voulu. C'est exactement l'analogie PDE de : « en linéaire, si $\dot{X} = AX$ croît comme $e^{\alpha t}$, choisir $\mu(t) = \mu_0 e^{\beta t}$ avec $\beta > \alpha$ garantit $|X|/\mu \rightarrow 0$. » :contentReference[oaicite :11]index=11

Lemme 4 (Décroissance (ultimate bound) sur un intervalle de durée T à μ fixé). *Supposons μ constant sur $(t_1^*, t_1^* + T]$ et que l'état au temps t_1^* soit dans une région « sûre » (non saturation + marge). Alors, sur $(t_1^*, t_1^* + T]$:*

$$\sup_{x \in [0, D]} e^{\frac{\nu \ln(1+\lambda)}{D}(x-D)} |w(t, x)| \leq \max \left\{ \text{niveau initial}, \Omega C M \mu \right\},$$

et en particulier au temps $t_1^* + T$ on obtient un rétrécissement compatible avec une mise à jour $\mu \leftarrow \Omega \mu$.

Idée (preuve). L'article réécrit la dynamique cible en une équation intégrale de type *retard* (delay) et utilise un fonctionnel de Lyapunov V_1 (pondération exponentielle sur $[-D, 0]$), puis une analyse ISS/small-gain pour borner l'effet du terme de quantification (erreur bornée $\Delta\mu$) et garantir une contraction sur une fenêtre T . :contentReference[oaicite :12]index=12

8.3 Théorème (stabilité globale asymptotique)

Théorème 2 (Structure du résultat). *Sous une condition explicite reliant la **précision relative** Δ/M et les constantes (M_1, M_2, M_3) (données par l'EDP et les transformations), il est possible de choisir $\lambda > 0$, $\nu \in (0, 1)$, $\Omega \in (0, 1)$ et $T > 0$ de sorte que la loi hybride (zoom-out puis zoom-in) assure*

$$\|u(t)\|_\infty \longrightarrow 0 \quad \text{quand } t \rightarrow \infty.$$

Lecture pédagogique (parallèle cas linéaire). C'est *exactement* le même mécanisme que dans votre cours linéaire :

- si μ est trop petit : saturation $\Rightarrow$ instabilité (Fig. 3 de l'article) ;
- si μ est trop grand : pas de saturation mais erreur $\Delta\mu$ énorme $\Rightarrow$ pas de convergence (Fig. 4 : cycles/plateaux) ;
- $\Rightarrow$ stratégie hybride : **capturer** (éviter saturation) puis **dézoomer** (réduire erreur) par paliers garantissant une contraction.

Les Figures 3–4 discutées dans l'article illustrent exactement ces deux écueils et motivent la loi hybride. :contentReference[oaicite :13]index=13 :contentReference[oaicite :14]index=14

9 (B) Quantification sur l'entrée : $U(t) = q_\mu(U_{\text{nom}}(t))$

Vous avez demandé explicitement ce cas : **quantification sur l'entrée**

$$U(t) = q_\mu(U_{\text{nom}}(t)) \quad \text{avec} \quad U_{\text{nom}}(t) = \int_0^D k(D, x)u(t, x) dx$$

(backstepping nominal), et une loi hybride pour μ .

9.1 Loi hybride (article, Section 4)

Quand l'état est disponible (non quantifié), mais l'entrée est quantifiée, l'article propose :

$$U(t) = \begin{cases} 0, & 0 \leq t \leq \bar{t}_1^*, \\ \mu(t) q\left(\frac{1}{\mu(t)} \int_0^D k(D, x)u(t, x) dx\right), & t > \bar{t}_1^*, \end{cases} \quad (\text{IQ})$$

où la stratégie $\mu(t)$ est de même forme que pour la quantification d'état, et $\bar{t}_1^*$ est le *premier instant* où l'événement de capture est vérifié : :contentReference[oaicite :15]index=15

$$\|u(\cdot, \bar{t}_1^*)\|_\infty \leq \frac{M}{(1 + \lambda)^\nu} \frac{M_2}{M_1 M_3} \mu(\bar{t}_1^*). \quad (\text{Cap-IQ})$$

Cet événement est *délectable* ici puisque u est mesuré (non quantifié).

9.2 Structure de preuve (analogue au cas quantification d'état)

Lemme 5 (Capture (entrée quantifiée)). *Il existe $\bar{t}_1^* < \infty$ tel que (Cap-IQ) est satisfaite.*

Idée. Sur $[0, \bar{t}_1^*]$, on est encore en boucle ouverte ($U \equiv 0$), et le raisonnement est identique : on choisit $\mu(t)$ croissante pour « rattraper » la croissance de l'état, puis on déclenche quand $\|u\|_\infty/\mu$ est suffisamment petit. :contentReference[oaicite :16]index=16 :contentReference[oaicite :17]index=17

Lemme 6 (Décroissance sur une fenêtre T (entrée quantifiée)). *Sous une condition explicite sur Δ/M (similaire à celle du cas mesures quantifiées), on obtient une borne de type ISS/ultimate-bound sur la dynamique transformée, qui permet de diminuer μ par paliers $\mu \leftarrow \Omega\mu$ et donc d'assurer la convergence.*

Théorème 3 (Stabilité globale asymptotique (entrée quantifiée)). *Sous la condition de Theorem 2 (Section 4) de l'article, le schéma hybride (IQ) garantit $\|u(t)\|_\infty \rightarrow 0$.*

Remarque 9 (Pourquoi ce cas est souvent « plus simple » en pratique). En entrée quantifiée, l'erreur apparaît comme une perturbation au bord (via U), et le nominal backstepping est typiquement ISS vis-à-vis d'une perturbation de bord, ce qui facilite le small-gain. L'article compare explicitement la « permissivité » des conditions entre Theorem 1 et Theorem 2. :contentReference[oaicite :18]index=18

10 Quantification de sortie : recette (observer/backstepping) + parallèles

Pourquoi c'est un cran au-dessus. Dans le papier SCL fourni, les deux cadres traités sont :

- mesures d'état quantifiées (state quantization) ;
- entrée quantifiée (input quantization) avec état disponible.

La **quantification de sortie** (par exemple une seule mesure de bord $y(t) = u(t, 0)$ ou $y(t) = u(t, D)$, quantifiée) demande en général un **observateur** (souvent backstepping observer) avant de réinjecter un retour.

Architecture standard (pédagogique).

$$\boxed{\text{Sortie quantifiée } y_q(t) = q_\mu(y(t))} \implies \boxed{\text{Observateur PDE } \hat{u}} \implies \boxed{U(t) = \text{retour backstepping basé sur } \hat{u}}$$

Puis on reproduit le même raisonnement :

- établir des estimations ISS pour l'erreur d'observation (entrée : erreur de quantification sur la mesure) ;
- coupler (small-gain) observateur + boucle de commande ;
- introduire $\mu(t)$ pour éviter saturation et réduire erreur.

Remarque 10 (Ce que je vous propose pour la suite). Si vous me dites quelle **sortie** vous visez (mesure de bord ? mesure ponctuelle ? moyenne spatiale ?), je vous fournis la version complète (observateur backstepping + preuve) avec un choix concret. Ici, je vous ai donné la « recette » cohérente avec votre logique et le papier SCL.

11 Simulation : reproduire les figures (norme, cartes, $U(t)$, $\mu(t)$) en Python

11.1 Ce qu'on vise (proche des Figures 3–5)

L'article illustre (au moins) :

- $\|u(t)\|_\infty$ et une carte $u(t, x)$ pour μ fixé petit (instable car saturation) ;

- $\|u(t)\|_\infty$ et $u(t, x)$ pour μ fixé grand (borné mais pas convergent : erreur grande) ;
- le cas hybride (ou le cas entrée quantifiée) qui converge. :contentReference[oaicite :19]index=19

11.2 Choix numériques

- Domaine : $x \in [0, D]$ avec $D = 1$.
- Temps : $t \in [0, T_{\max}]$.
- Schéma : *upwind* pour transport (CFL).
- Paramètres quantifieur : $M = 2$, $\Delta = M/40$ (comme en figures). :contentReference[oaicite :21]index=21
- Exemple initial : un profil suffisamment grand pour provoquer saturation si μ est petit.

11.3 Code Python (un fichier) : `transport_quantized_control.py`

Instructions. Copiez-collez ce code dans un fichier `transport_quantized_control.py` et exécutez-le. Il produit :

- Figure A : $\|u(t)\|_\infty$ (et $\mu(t)$ sur le même graphe si souhaité) ;
- Figure B : carte $u(t, x)$ (heatmap) ;
- Figure C : $U(t)$ (nominal vs quantifié si vous activez l’option).

Remarque importante : Le code ci-dessous est volontairement *pédagogique* et modulaire (vous pourrez remplacer facilement la loi $\mu(t)$ par exactement celle de votre implémentation actuelle). Les noyaux $k(D, x)$ sont implémentés dans le cas constant (celui des figures). :contentReference[oaicite :22]index=22

À ajuster pour coller *exactement* au papier. Pour reproduire au plus près les figures (incluant la carte $u(t, x)$ à droite), il faut :

- stocker $u(t, \cdot)$ à chaque pas (ou chaque N_s pas) et tracer une heatmap ;
- remplacer la loi « pédagogique » de $\mu(t)$ par la loi *piecewise-constante* et les temps T, τ, Ω tels que définis dans l’article (Section 3 pour état quantifié, Section 4 pour entrée quantifiée) ; :contentReference[oaicite :23]index=23
- utiliser les constantes M_1, M_2, M_3 et les paramètres (λ, ν) pour un choix *constructif* de T garantissant la contraction (c’est ce que fait Lemma 2 via V_1). :contentReference[oaicite :24]index=24

Références

- [1] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems : Theory and Design*. Prentice Hall, 3 ed., 1997.
- [2] R. M. Gray and D. L. Neuhoff, “Quantization,” *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, 1998.
- [3] D. Liberzon, “Hybrid feedback stabilization of systems with quantized signals,” *Automatica*, vol. 39, no. 9, pp. 1543–1554, 2003.
- [4] R. W. Brockett and D. Liberzon, “Quantized feedback stabilization of linear systems,” *IEEE Transactions on Automatic Control*, vol. 45, no. 7, pp. 1279–1289, 2000.
- [5] J. Cai, M. Takemoto, and H. Nakajo, “A deep look into logarithmic quantization of model parameters in neural networks,” in *Proceedings of the 10th International Conference on Advances in Information Technology (IAIT)*, pp. 1–8, 2018.

- [6] S. Oh, H. Sim, S. Lee, and J. Lee, “Automated log-scale quantization for low-cost deep neural networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 742–751, 2021.
- [7] H. K. Khalil, *Nonlinear Systems*. Prentice Hall, 3 ed., 2002.