Le génie pour l'industrie

Présentation mi-projet : Contrôle de la trajectoire d'une voiture de course

Étudiant : Léon JUBIN-HUERNE
Cours : SYS802 : Commandes avancées
Professeur : David Bensoussan

1 novembre 2022

Sommaire

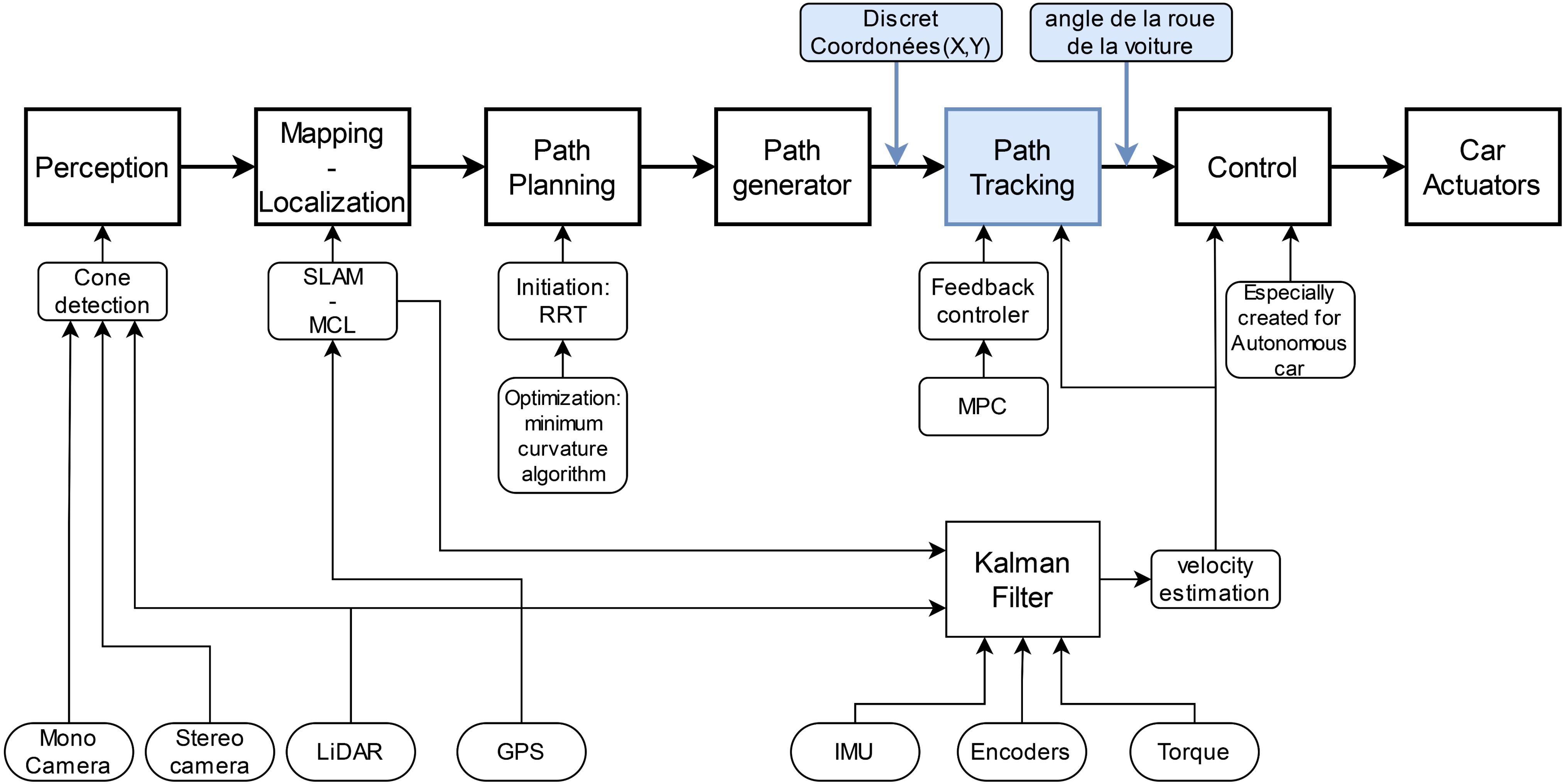
1. Introduction
    1. Contexte
    2. Stratégie autonome
    3. Problématique de contrôle de trajectoire
2. Présentation du modèle
3. Méthode de contrôle
    1. Méthode géométrique : Pure Pursuite
    2. Méthode boite noir : PID
4. Schémas-bloc
5. Simulation Matlab

1.1. Contexte : Formule ÉTS - voiture autonome

Formule ETS à Toronto Shoot-Out
Voiture vainqueur

Formule UAS Munich autonome
Épreuve de Skidpad

1.2. Stratégie autonome



1.3. Problématique : Schéma global

On cherche à trouver une commande pour faire le suivie d'une trajectoire dans le cadre d'une voiture de course autour d'un circuit.

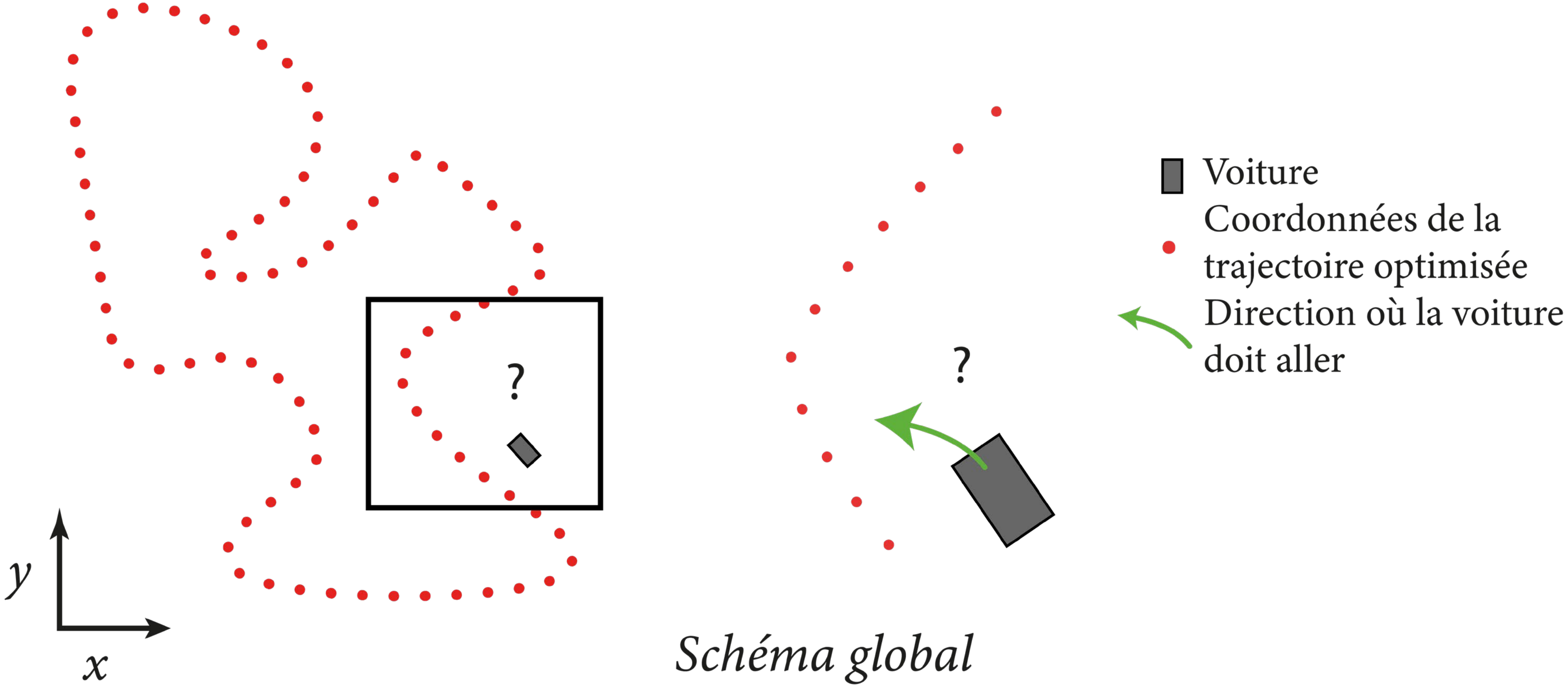
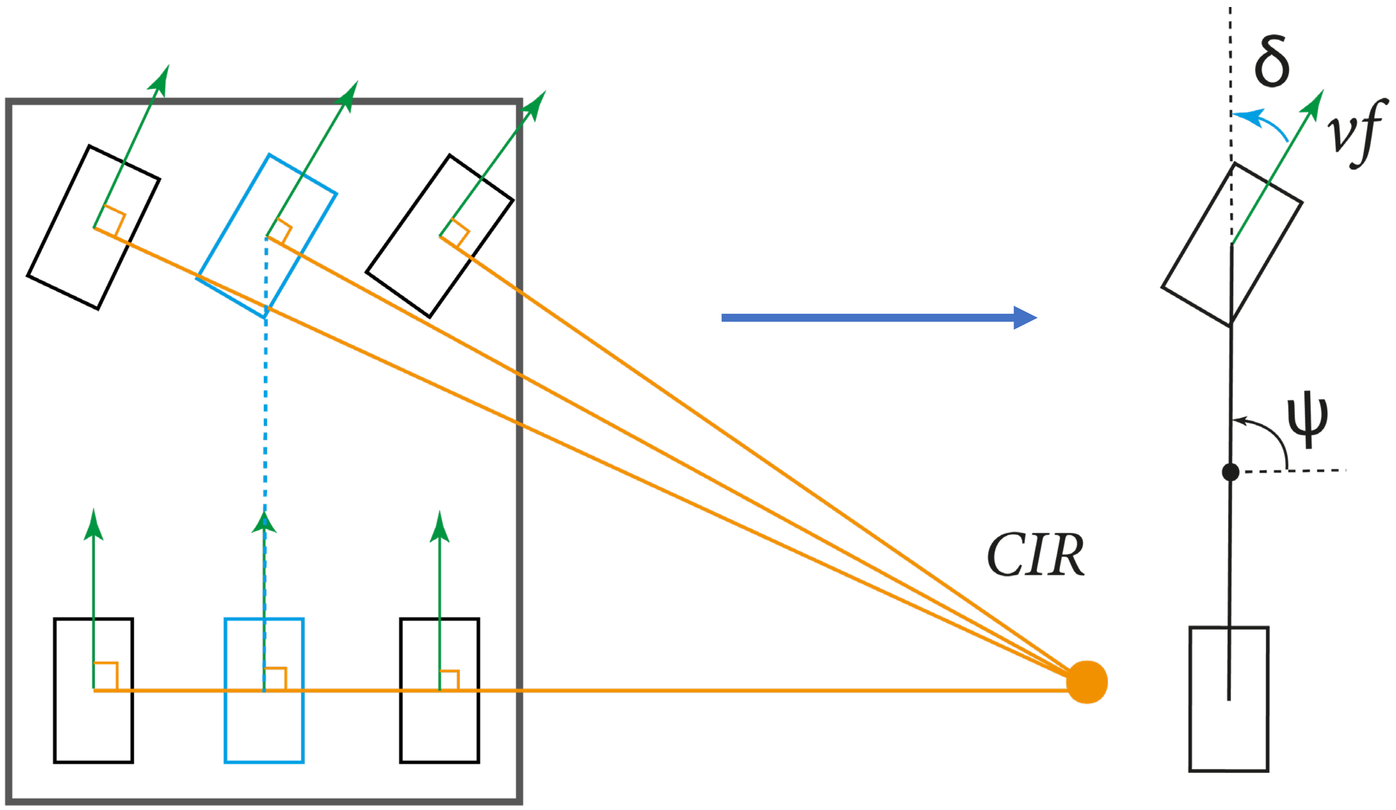


Schéma global

2. Présentation du modèle : bicycle model

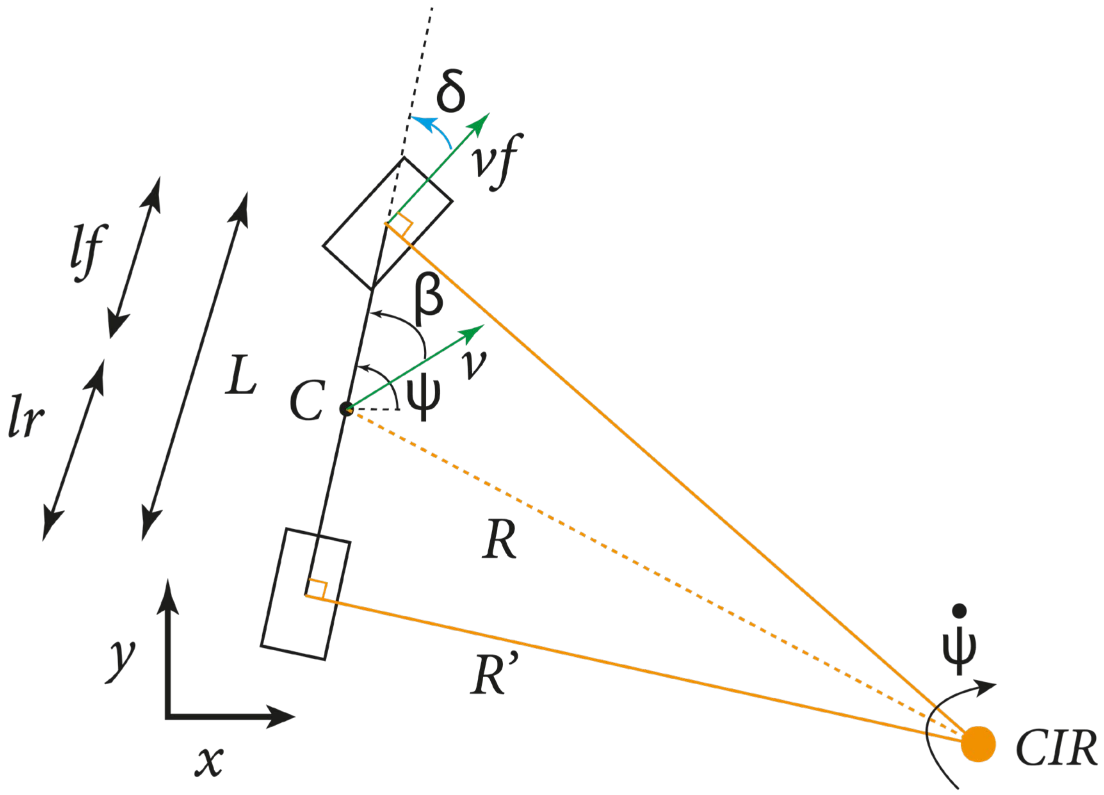


Modele d'Ackermann

Bicycle model

δ : angle de la roue avant
vf : vitesse de la roue avant
ψ : lacet (Yaw)
CIR : Centre Instantané de Rotation

2. Cinématique du bicycle model - Schéma



Bicycle model en cinématique

δ : angle de la roue avant
v : vitesse du véhicule
vf : vitesse de la roue avant
ψ : lacet (Yaw)
ψ : vitesse de rotation
β : slip angle
C : centre de gravité
L : distance inter roue ($lf + lr = L$)
lf : distance avec le Cet la roue avant
lr : distance avec le C et la roue arriere
R : distance en C et le CIR
R' : distance entre la roue arriere et le CIR
CIR : Centre Instantané de Rotation

2. Cinématique du bicycle model – Équations

$$\text{Bicycle model temporel :} \begin{cases} \dot{x} = v \cdot \cos(\psi + \beta) \\ \dot{y} = v \cdot \sin(\psi + \beta) \\ \dot{\psi} = \dfrac{v}{l_r + l_f} \cdot \cos\beta \cdot \tan\delta \end{cases} \quad avec\ \beta = \arctan\left(\frac{l_r}{l_r + l_f} \cdot \tan\delta\right)$$

Pas linéaire !

Petit angle

$$\text{Bicycle model temporel simplifié :} \begin{cases} \dot{x} = v \cdot \cos(\psi) \\ \dot{y} = v \cdot \sin(\psi) \\ \dot{\psi} = \dfrac{v}{L} \cdot \delta \end{cases}$$

Pas linéaire !

Modèle de la voiture

Développement limité ordre 1

Passage en z

$$\text{Bicycle model discret linéarisé :} \begin{cases} x(z) = \dfrac{T_e \cdot v}{1 - z^{-1}} \\ y(z) = \dfrac{T_e \cdot v \cdot \psi(z)}{1 - z^{-1}} \\ \psi(z) = \dfrac{T_e \cdot v}{L(1 - z^1)} \cdot \delta(z) \end{cases}$$

Linéaire !

Cinématique du bicycle model - Jacobien

J'ai les matrices du jacobien, ce n'est pas concluant pour l'instant :

$$A = \begin{pmatrix} 0 & 0 & -v\,\sin\left(\psi\left(t\right)\right) & 0 \\ 0 & 0 & v\,\cos\left(\psi\left(t\right)\right) & 0 \\ 0 & 0 & 0 & \frac{v}{L} \\ 0 & 0 & 0 & 0 \end{pmatrix} \sim \begin{pmatrix} 0 & 0 & -v\,\psi\left(t\right) & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 0 & \frac{v}{L} \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ et } \quad B = \begin{pmatrix} 0 \\ 0 \\ \frac{v}{L} \\ 0 \end{pmatrix}$$

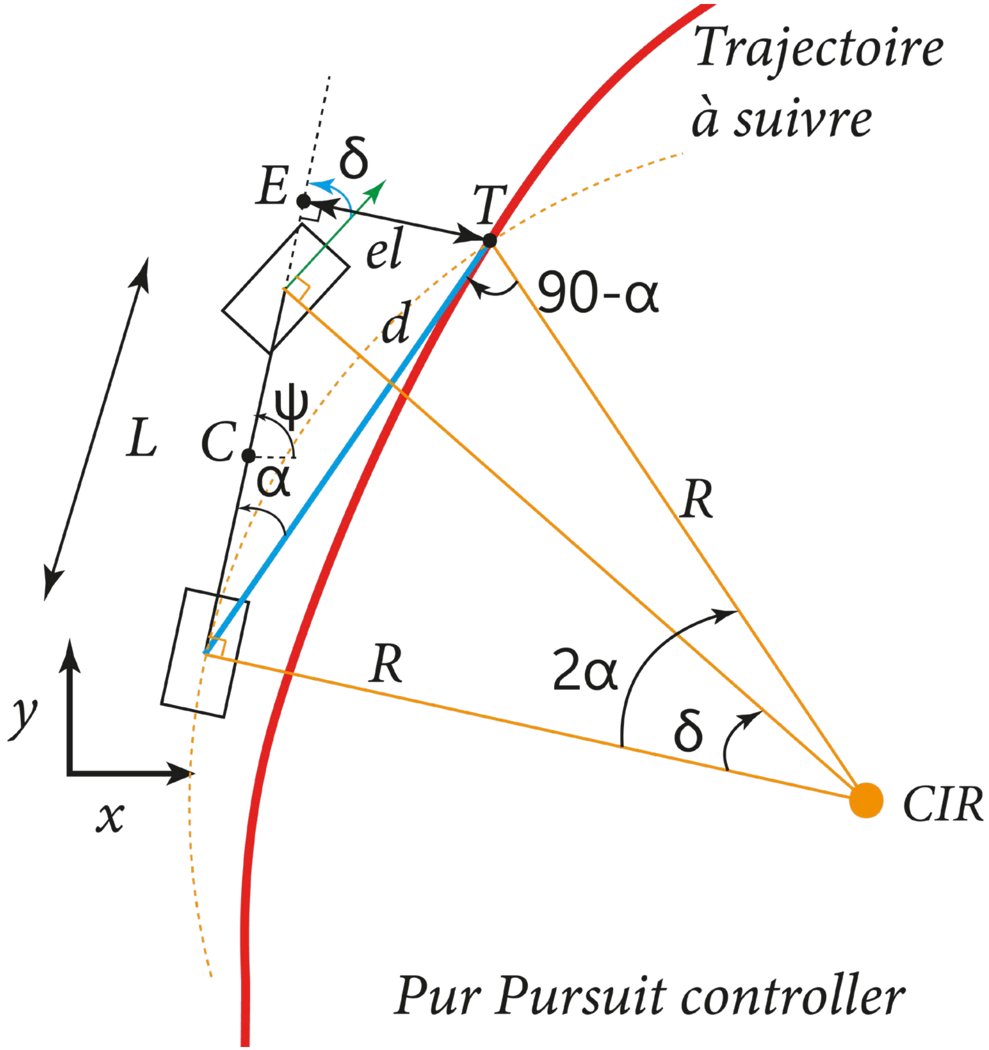
Ainsi, grâce au jacobien on trouve que $\dot{X}$:

$$\dot{X} = \dot{X}_0 + \Delta\dot{X} = \begin{pmatrix} x_0 & x_0 & x_0 - \Delta x \cdot v\sin(\psi) & x_0 \\ y_0 & y_0 & y_0 + \Delta y \cdot v\cos(\psi) & y_0 \\ \psi_0 + \frac{\Delta\delta \cdot v}{L} & \psi_0 + \frac{\Delta\delta \cdot v}{L} & \psi_0 + \frac{\Delta\delta \cdot v}{L} & \psi_0 + \frac{\Delta\delta \cdot v}{L} + \frac{\Delta\psi \cdot v}{L} \\ \delta_0 & \delta_0 & \delta_0 & \delta_0 \end{pmatrix}$$

2. Dynamique du bicycle model - Équations

À FAIRE

3.1. Méthode géométrique : Pure Pursuite - Schéma



Pur Pursuit controller

δ : angle de la roue avant
L : distance inter roue
R : rayon de braquage de la voiture
CIR : Centre Instantané de Rotation
T : point objectif (Target)
E : point erreur
el : erreur lateral entre la voiture et l'objectif
d : distance entre la roue arriere et l'objectif (look ahead distance)
ψ : lacet (Yaw)
α : angle entre la voiture et l'objectif

3.1. Méthode géométrique : Pure Pursuite - Équations

$$d(E,T) = e_l = \sqrt{[x_T - x_C + (\frac{L}{2} - d)\sin(\psi)]^2 + [y_T - y_C + (d - \frac{L}{2})\cos(\psi)]^2}$$

Pas linéaire !

Contrôleur

$$\delta = \frac{2L}{d^2} \cdot e_l$$

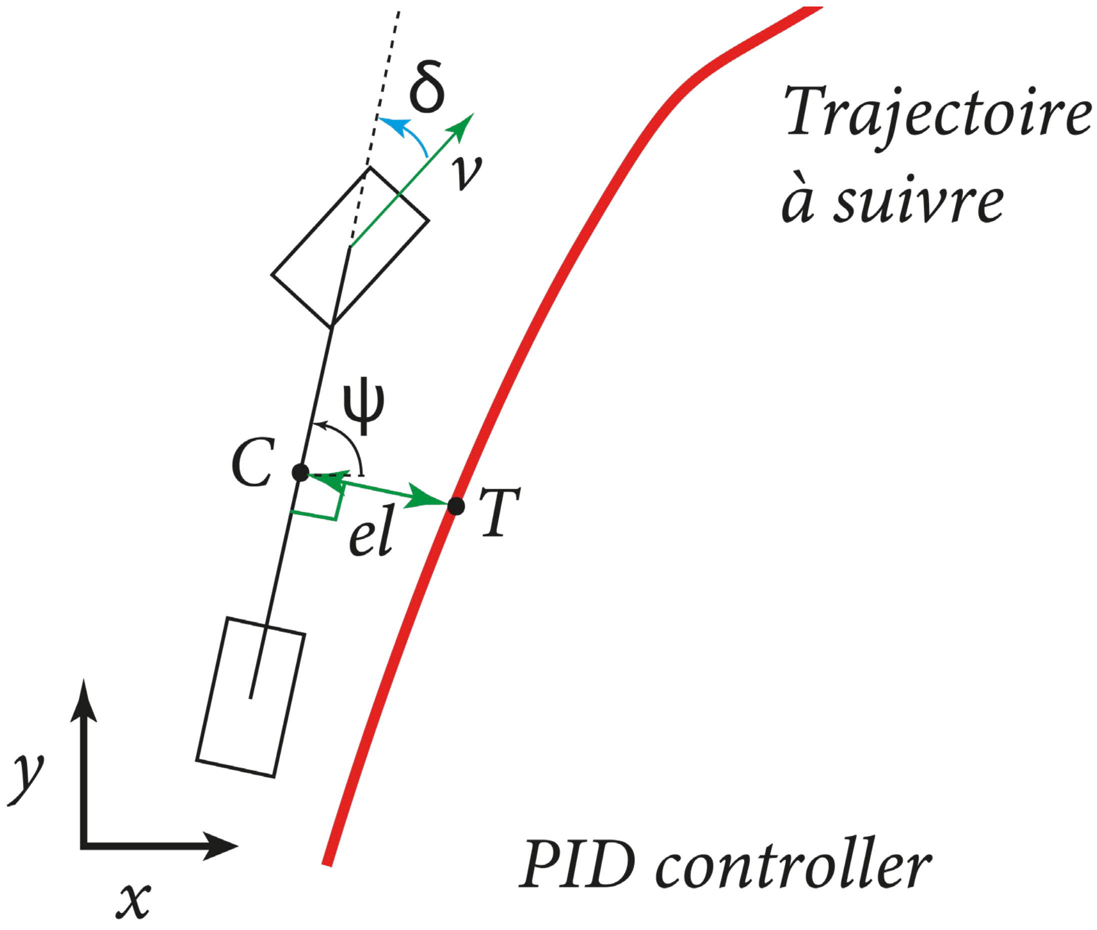
Développement limité ordre 1

Passage en z

Différentiel – Calcule de l’erreur

$$e_l(z) = x_g(z) - z^{-}1 \cdot x_c(z) + y_g(z) - z^{-}1 \cdot y_c(z) + z^{-}1 \cdot (\frac{L}{2} - d) \cdot \psi(z) - (0.5 + \frac{L}{2} - d)$$

3.2 Méthode boîte noire : PID - Schéma



δ : angle de la roue avant
v : vitesse du véhicule
C : Centre de gravité
e : distance entre C et la trajectoire à suivre
ψ : lacet (Yaw)
T : point objectif (Target)

3.2 Méthode boîte noire : PID - Équations

Contrôleur

Le PID en z :

$$\delta(z) = K_p \cdot e(z) + K_i \cdot \frac{Te \cdot z}{z-1} + K_d \cdot \frac{z-1}{Te \cdot z}$$

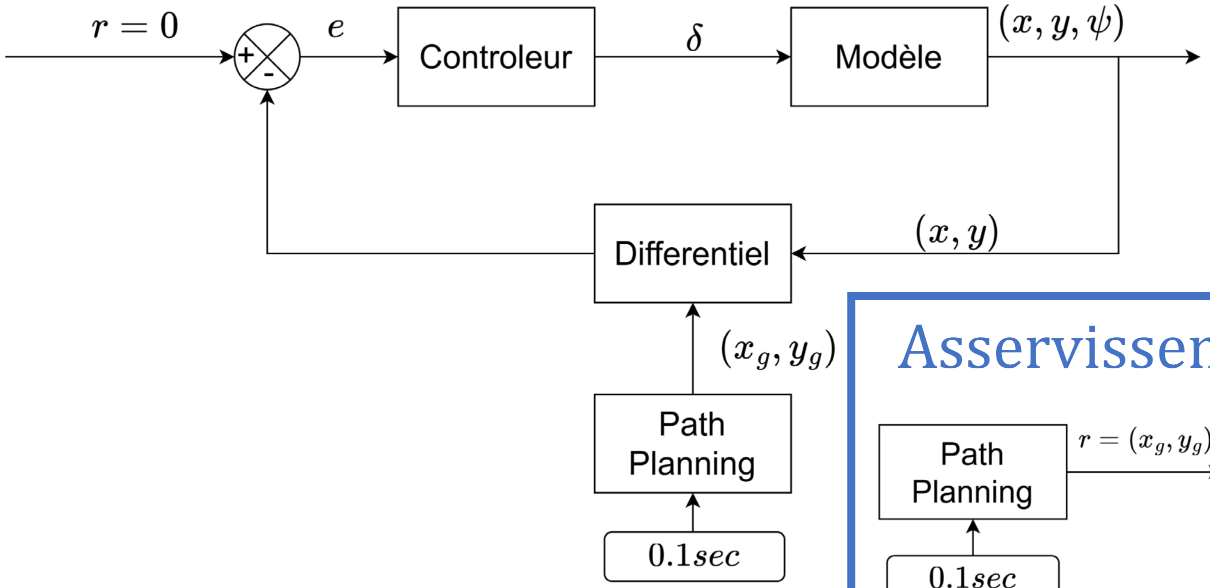
Différentiel – Calcule de l’erreur

$$d(T(k), C(k)) = e_l(k) = \sqrt{(x_T(k) - x_C(k-1))^2 + (y_T(k) - y_C(k-1))^2}$$

Pas linéaire ! => À FAIRE : linéarisation (DL1) et passage en Z

Schéma – bloc : régulation ou asservissement ?

Régulation



Asservissement

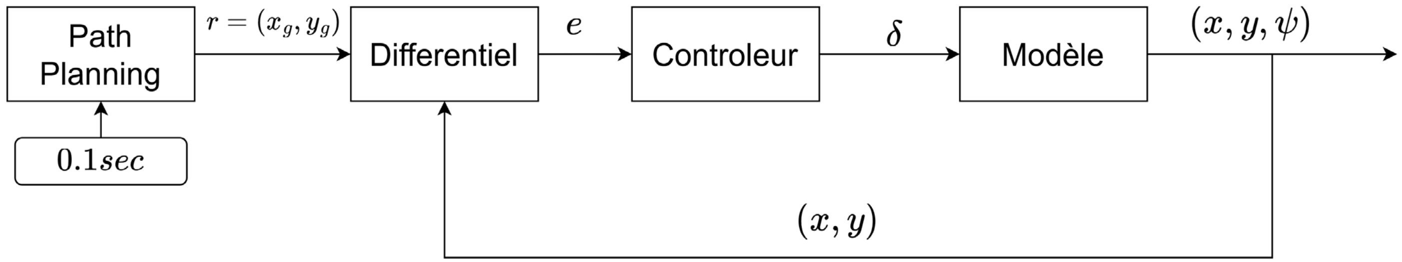
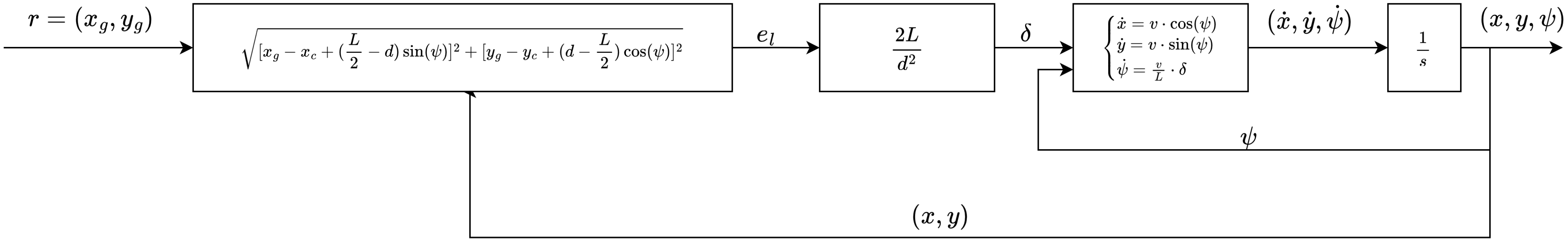
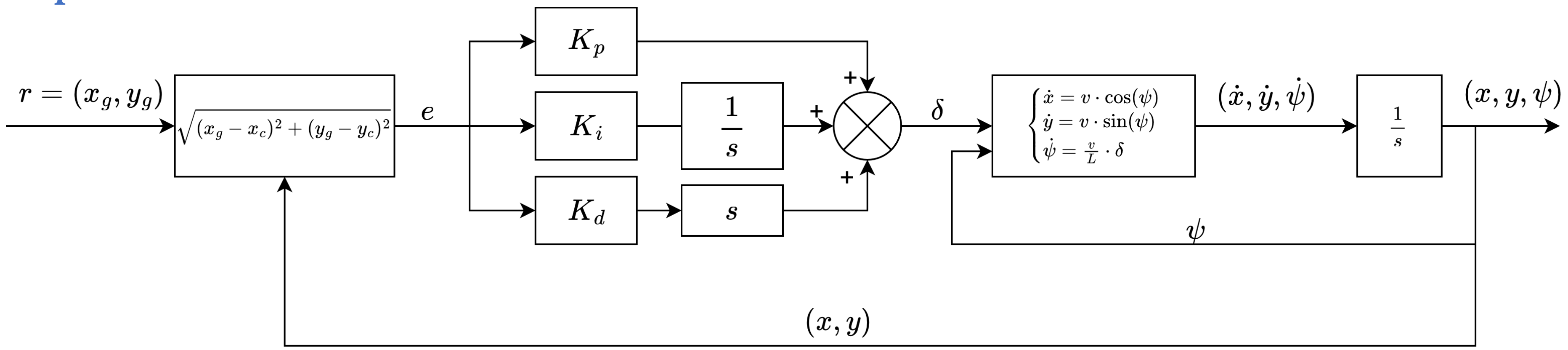


Schéma – bloc : Pure Pursuite / PID

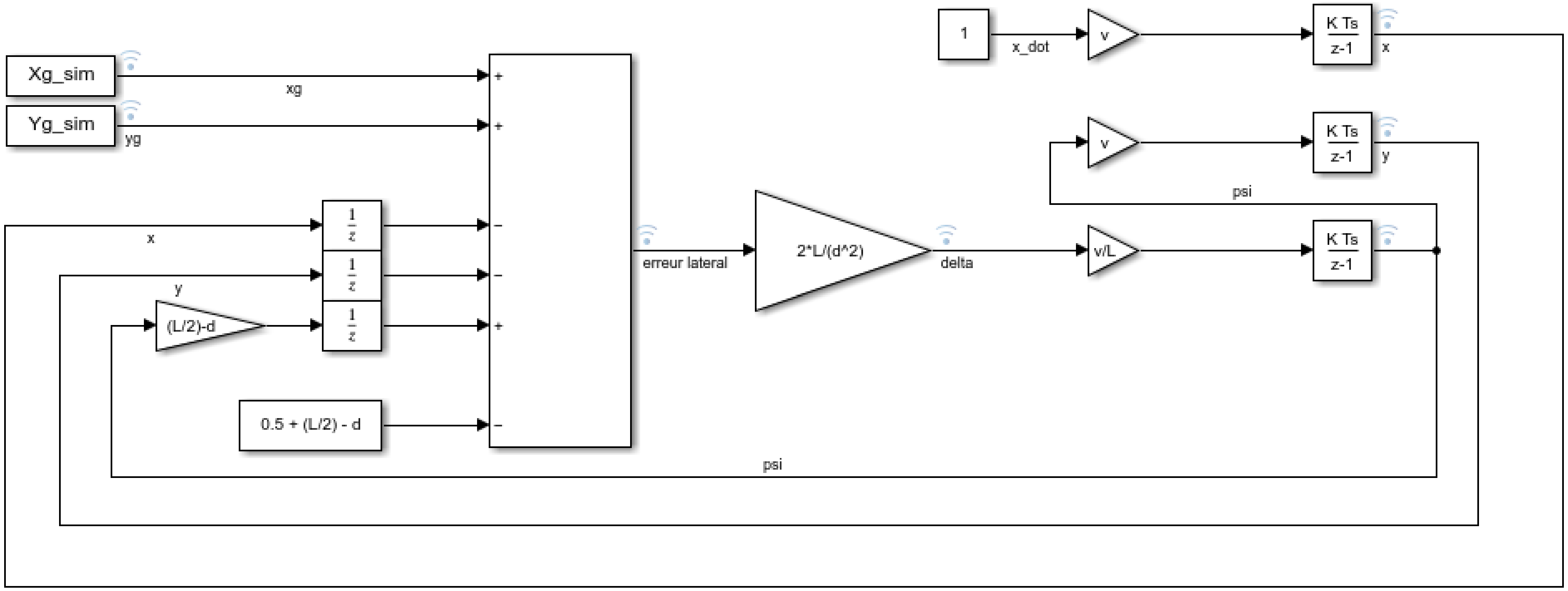
Pure Pursuite temporel



PID temporel



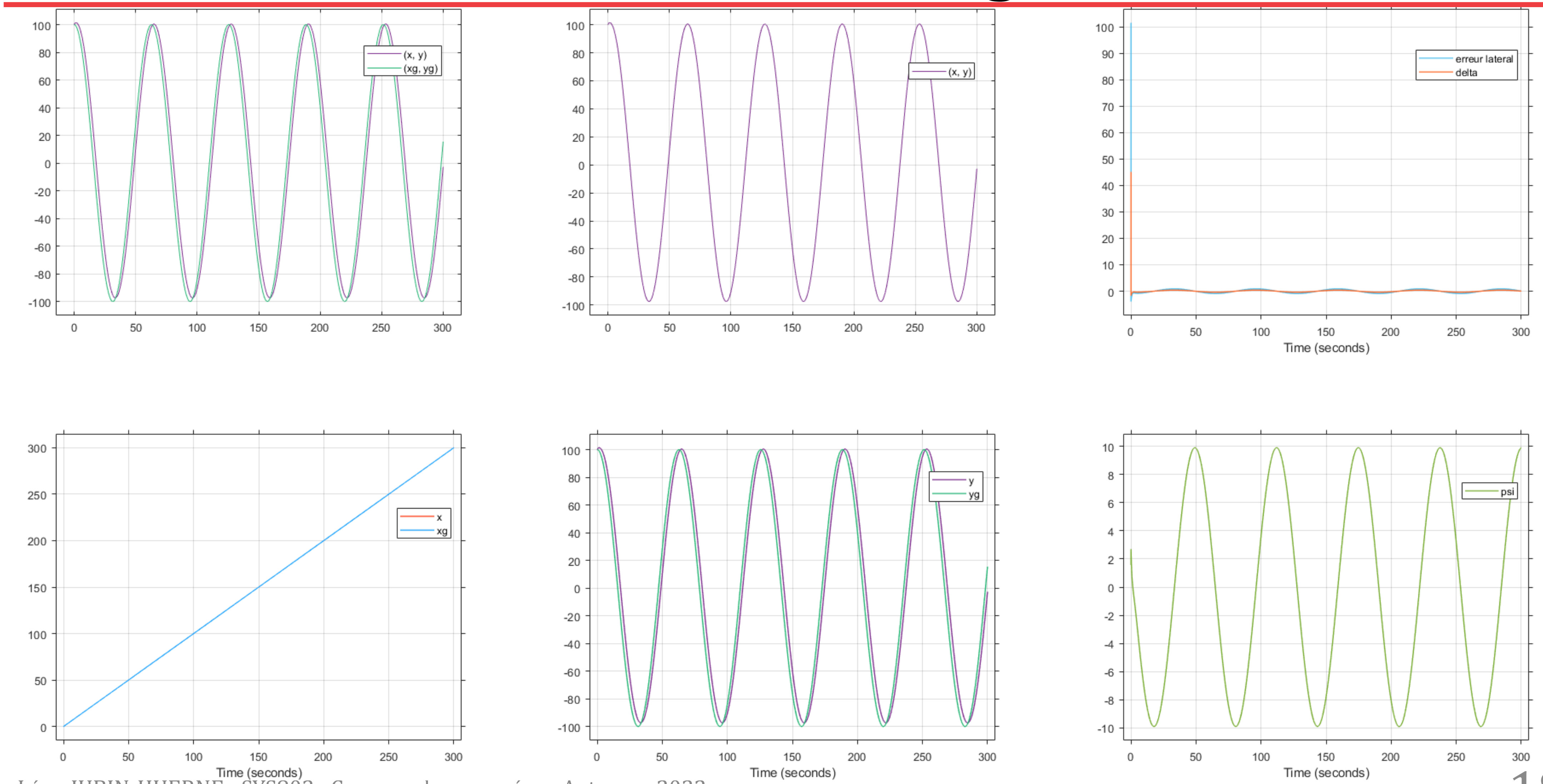
Simulation Matlab : Simulink Pure Pursuite



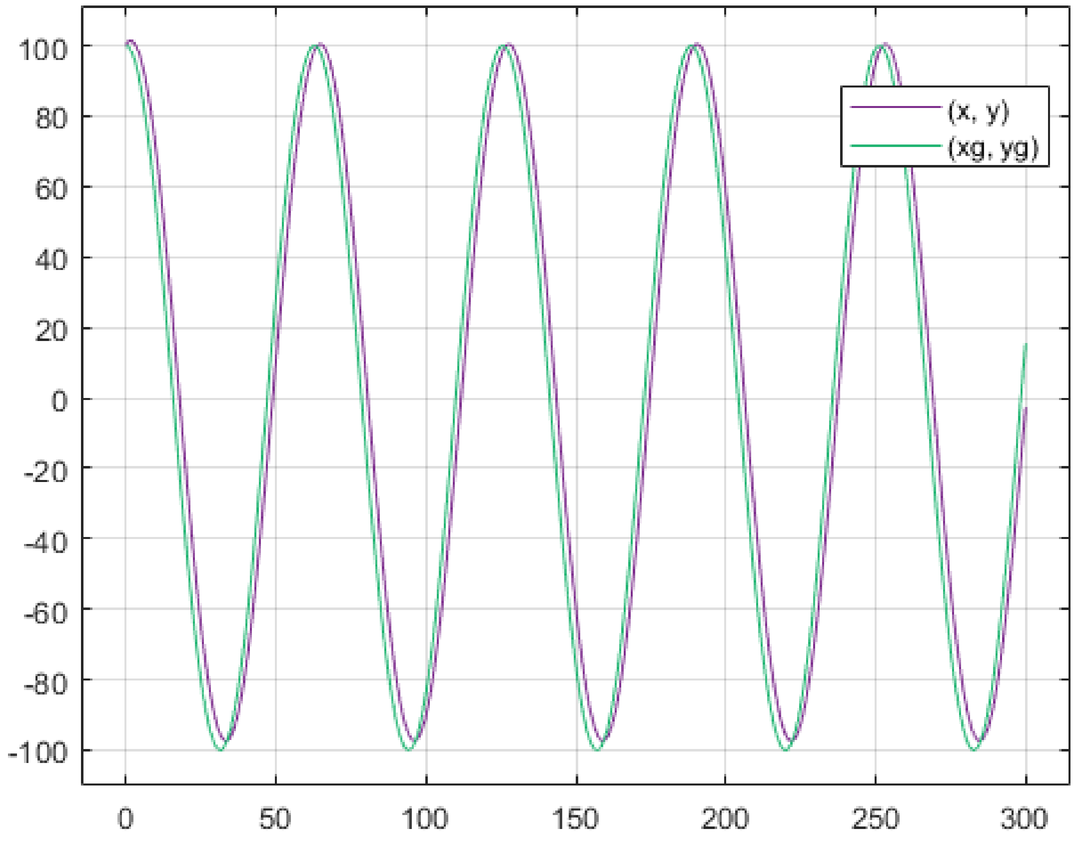
<u>Conditions initiales</u> :

- d = 3 m
- v = 5 m/s
- L = 2 m
- Psi = 90°
- (x,y) = (0,0)

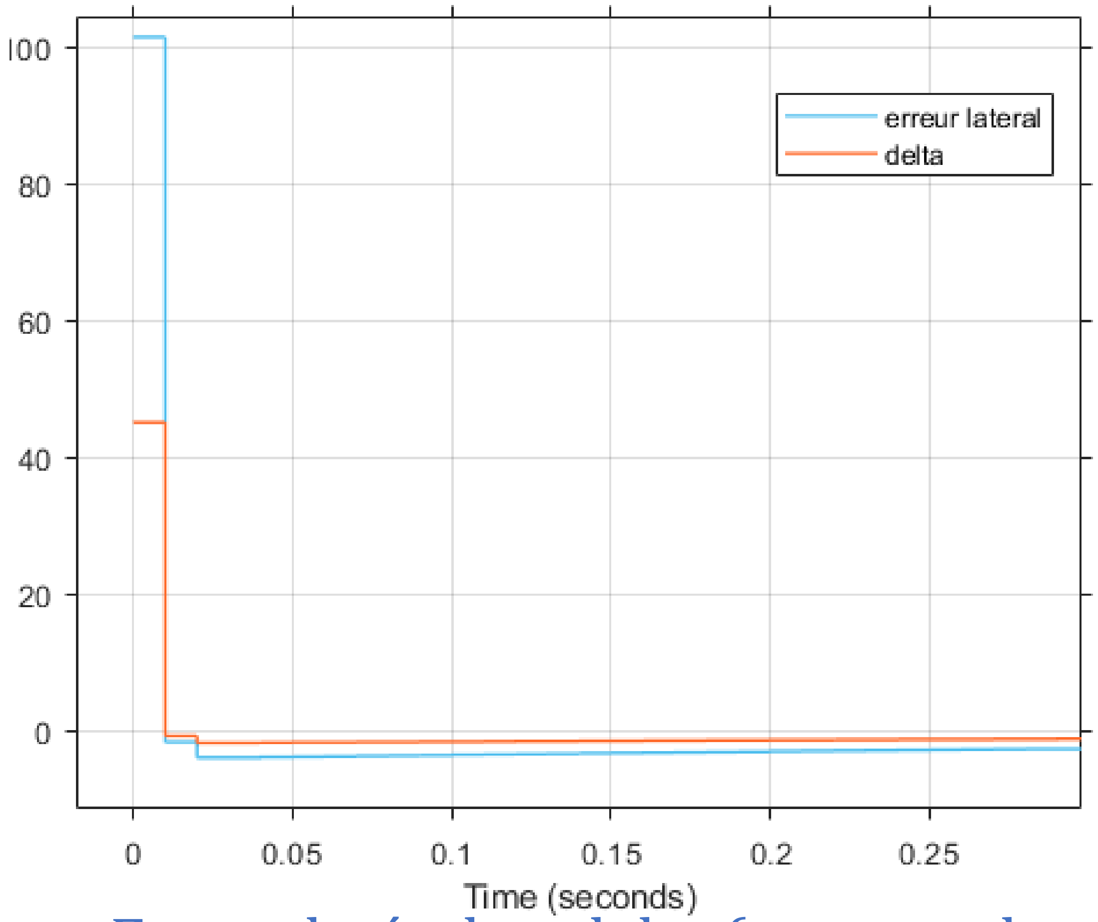
Simulation Matlab : Résultats globaux



Simulation Matlab : Résultats détaillés



Référence (xg, yg) – Sortie (x, y)



Erreur latérale – delta (commande du système)

Simulation Matlab : Simulink PID

À FAIRE

Conclusion

- <u>Résultat :</u>
  - On peut voir que la commande linéarisé pour le système cinématique est précise pour le Pure Pursuite (le gap est dû au paramètre **d**) (ça marche à une arnaque près)

- <u>Ce qu'il reste à faire :</u>
  - Avec le modèle cinématique :
    - Comprendre l'arnaque du Pur Pursuite
    - Référence pour le Pure Pursuite plus compliqué (angle droit) et plus représentatif (boucle, circuit (circuit de Toronto / circuit skidpad / circuit endurance Michigan))
    - Pure Pursuite : d variable ? d qui dépend de la vitesse ?
    - Faire la même chose pour le PID
    - Faire test du code avec les paramètres choisis avec les fonctions non linéaires
    - Vitesse variable ?
  - Développer la même méthode pour le modèle dynamique

Merci pour votre attention !

Questions ?

Étudiant : Léon JUBIN-HUERNE
Cours : SYS802 : Commandes avancées
Professeur : David Bensoussan

1 novembre 2022

Références

- Ar, K. B. (s.d.). IDENTIFICATION OF LINEAR BICYCLE MODEL OF AN AUTOMOBILE USING EXPERIMENTAL DATA, 9.
- Betz, J., Zheng, H., Liniger, A., Rosolia, U., Karle, P., Behl, M., … Mangharam, R. (2022). Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing. *IEEE Open Journal of Intelligent Transportation Systems*, *3*, 458-488. https://doi.org/10.1109/ojits.2022.3181510
- Chen, Y., Shan, Y., Chen, L., Huang, K., & Cao, D. (2018). Optimization of Pure Pursuit Controller based on PID Controller and Low-pass Filter. Dans *2018 21st International Conference on Intelligent Transportation Systems (ITSC)* (pp. 3294-3299). Maui, HI : IEEE. https://doi.org/10.1109/ITSC.2018.8569416
- Corke, P. (2017). *Robotics, Vision and Control* (Vol. 118). Cham : Springer International Publishing. https://doi.org/10.1007/978-3-319-54413-7
- De Luca, A., Oriolo, G., & Samson, C. (1998). Feedback control of a nonholonomic car-like robot. Dans J.-P. Laumond (Éd.), *Robot Motion Planning and Control* (Vol. 229, pp. 171-253). Berlin, Heidelberg : Springer Berlin Heidelberg. https://doi.org/10.1007/BFb0036073
- Kabzan, J., Valls, M. de la I., Reijgwart, V., Hendrikx, H. F. C., Ehmke, C., Prajapat, M., … Siegwart, R. (2019, 13 mai). AMZ Driverless: The Full Autonomous Racing System. arXiv. Repéré à http://arxiv.org/abs/1905.05150
- Paden, B., Cap, M., Yong, S. Z., Yershov, D., & Frazzoli, E. (2016, 25 avril). A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles. arXiv. Repéré à http://arxiv.org/abs/1604.07446
- Rankin, A. L., Crane III, C. D., & Armstrong II, D. G. (1998). Evaluating a PID, pure pursuit, and weighted steering controller for an autonomous land vehicle. Dans D. W. Gage (Éd.), (pp. 1-12). Communication présentée au Intelligent Systems & Advanced Manufacturing, Pittsburgh, PA. https://doi.org/10.1117/12.299554
- Snider, J. M. (2009). *Automatic Steering Methods for Autonomous Automobile Path Tracking*. Robotics Institute Carnegie Mellon University Pittsburgh, Pennsylvania.
- Watzenig, D., & Horn, M. (Éds). (2017). *Automated Driving: Safer and More Efficient Future Driving*. Cham : Springer International Publishing. https://doi.org/10.1007/978-3-319-31895-0
- Yu, J.-H. (s.d.). *IMPLEMENTATION OF PATH TRACKING ALGORITHMS AND TRAJECTORY OPTIMIZATION BASED ON THE EXTENDED KALMAN FILTER*.