



# Robotique autonome

## Navigation

Francis Colas

# Introduction

## Planification

- espace de configuration et algorithmes de planification ;
- carte **connue** et **statique**.

# Introduction

## Planification

- espace de configuration et algorithmes de planification ;
- carte **connue** et **statique**.

## Navigation

- mouvement d'un robot mobile :
  - planification de trajectoire,
  - exécution de la trajectoire,
  - évitement d'obstacles ;
- exploration :
  - dans un environnement inconnu,
  - décider des mouvements pour construire la carte.

# Introduction

## Planification

- espace de configuration et algorithmes de planification ;
- carte connue et statique.

## Navigation

- mouvement d'un robot mobile :
  - planification de trajectoire,
  - exécution de la trajectoire,
  - évitement d'obstacles ;
- exploration :
  - dans un environnement inconnu,
  - décider des mouvements pour construire la carte.

## Objectif de la séance

- suivi de trajectoire ;
- évitement d'obstacle ;
- exploration.

# 1

## Suivi de trajectoire

## Suivi de trajectoire

### Suivi de trajectoire

- exécuter une trajectoire planifiée,
- à l'aide des capteurs,
- en prenant en compte les contraintes du robot ;
- correction des dérives.

# Suivi de trajectoire

## Suivi de trajectoire

- exécuter une trajectoire planifiée,
- à l'aide des capteurs,
- en prenant en compte les contraintes du robot ;
- correction des dérives.

## Principe

- trajectoire donnée ;
- position/erreur connue ;
- modèle de véhicule donné ;
- calcul d'une commande pour suivre la trajectoire.

# Suivi de trajectoire

## Suivi de trajectoire

- exécuter une trajectoire planifiée,
- à l'aide des capteurs,
- en prenant en compte les **contraintes** du robot ;
- correction des dérives.

## Principe

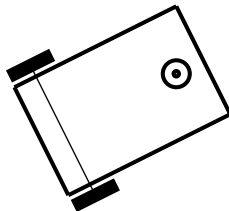
- trajectoire donnée ;
- position/erreur connue ;
- modèle de véhicule donné ;
- calcul d'une commande pour suivre la trajectoire.



## Modèles cinématiques

### Robot différentiel

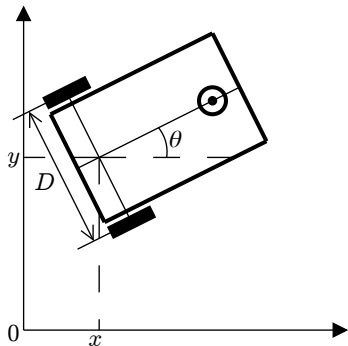
- deux roues motrices (droite et gauche) ;
- roue folle de stabilisation ;



# Modèles cinématiques

## Robot différentiel

- deux roues motrices (droite et gauche) ;
- roue folle de stabilisation ;
- configuration en position et orientation :  $(x, y, \theta)$  ;
- commande en vitesse des roues :  $(v_l, v_r)$  ;

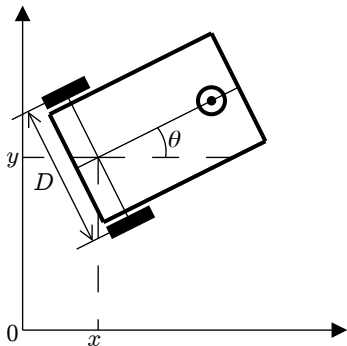


# Modèles cinématiques

## Robot différentiel

- deux roues motrices (droite et gauche) ;
- roue folle de stabilisation ;
- configuration en position et orientation :  $(x, y, \theta)$  ;
- commande en vitesse des roues :  $(v_l, v_r)$  ;
- modèle cinématique :

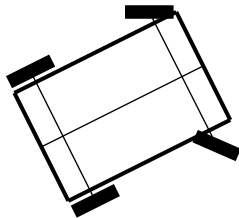
$$\begin{cases} \dot{x} &= \frac{v_r + v_l}{2} \cos \theta \\ \dot{y} &= \frac{v_r + v_l}{2} \sin \theta \\ \dot{\theta} &= \frac{v_r - v_l}{2D} \end{cases}$$



## Modèles cinématiques

### Voiture

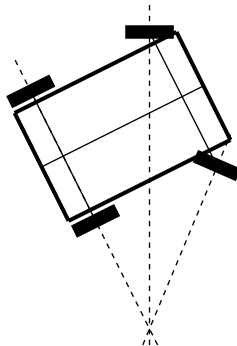
- roues avant orientables ;
- roues arrières fixes ;



# Modèles cinématiques

## Voiture

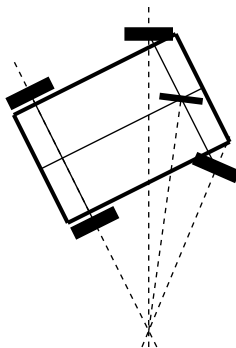
- roues avant orientables ;
- roues arrières fixes ;



# Modèles cinématiques

## Voiture

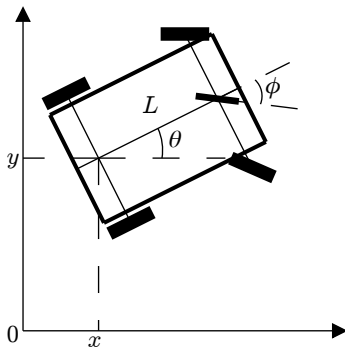
- roues avant orientables ;
- roues arrières fixes ;



# Modèles cinématiques

## Voiture

- roues avant orientables ;
- roues arrières fixes ;
- configuration en position, orientation et angle de braquage :  $(x, y, \theta, \phi)$  ;
- commande en vitesse et changement d'angle de braquage :  $(v, u)$ .

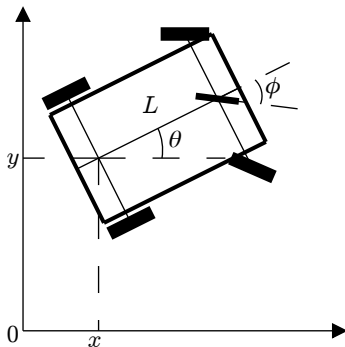


# Modèles cinématiques

## Voiture

- roues avant orientables ;
- roues arrières fixes ;
- configuration en position, orientation et angle de braquage :  $(x, y, \theta, \phi)$  ;
- commande en vitesse et changement d'angle de braquage :  $(v, u)$ .
- modèle cinématique :

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \frac{v}{L} \tan \phi \\ \dot{\phi} = u \end{cases}$$

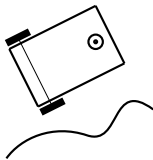




## Suivi de trajectoire

### Principe

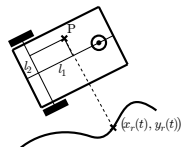
- définition des commandes en fonction de l'erreur ;
- obtention d'une équation différentielle sur l'erreur.



## Suivi de trajectoire

### Principe

- définition des commandes en fonction de l'erreur ;
- obtention d'une équation différentielle sur l'erreur.



### Robot différentiel

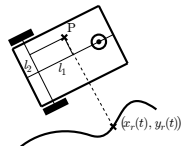
- mouvement d'un point :

$$\begin{pmatrix} \dot{x}_P \\ \dot{y}_P \end{pmatrix} = \begin{pmatrix} \frac{v_r + v_l}{2} \cos \theta - \frac{v_r - v_l}{2D} (l_1 \sin \theta + l_2 \cos \theta) \\ \frac{v_r + v_l}{2} \sin \theta - \frac{v_r - v_l}{2D} (-l_1 \cos \theta + l_2 \sin \theta) \end{pmatrix} = \mathbf{M} \begin{pmatrix} v_r \\ v_l \end{pmatrix}$$

## Suivi de trajectoire

### Principe

- définition des commandes en fonction de l'erreur ;
- obtention d'une équation différentielle sur l'erreur.



### Robot différentiel

- mouvement d'un point :

$$\begin{pmatrix} \dot{x}_P \\ \dot{y}_P \end{pmatrix} = \begin{pmatrix} \frac{v_r + v_l}{2} \cos \theta - \frac{v_r - v_l}{2D} (l_1 \sin \theta + l_2 \cos \theta) \\ \frac{v_r + v_l}{2} \sin \theta - \frac{v_r - v_l}{2D} (-l_1 \cos \theta + l_2 \sin \theta) \end{pmatrix} = \mathbf{M} \begin{pmatrix} v_r \\ v_l \end{pmatrix}$$

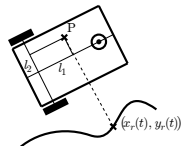
- erreur par rapport à la référence  $\mathbf{e} = (x_P - x_r(t), y_P - y_r(t))$  :

$$\dot{\mathbf{e}} = \mathbf{M} \begin{pmatrix} v_r \\ v_l \end{pmatrix} - \begin{pmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \end{pmatrix}$$

## Suivi de trajectoire

### Principe

- définition des commandes en fonction de l'erreur ;
- obtention d'une équation différentielle sur l'erreur.



### Robot différentiel

- mouvement d'un point :

$$\begin{pmatrix} \dot{x}_P \\ \dot{y}_P \end{pmatrix} = \begin{pmatrix} \frac{v_r + v_l}{2} \cos \theta - \frac{v_r - v_l}{2D} (l_1 \sin \theta + l_2 \cos \theta) \\ \frac{v_r + v_l}{2} \sin \theta - \frac{v_r - v_l}{2D} (-l_1 \cos \theta + l_2 \sin \theta) \end{pmatrix} = \mathbf{M} \begin{pmatrix} v_r \\ v_l \end{pmatrix}$$

- erreur par rapport à la référence  $\mathbf{e} = (x_P - x_r(t), y_P - y_r(t))$  :

$$\dot{\mathbf{e}} = \mathbf{M} \begin{pmatrix} v_r \\ v_l \end{pmatrix} - \begin{pmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \end{pmatrix}$$

- correction proportionnelle et pré-compensation :

$$\mathbf{M} \begin{pmatrix} v_r \\ v_l \end{pmatrix} = \begin{pmatrix} \dot{x}_r(t) \\ \dot{y}_r(t) \end{pmatrix} - \mathbf{K}\mathbf{e}$$

## Conclusion sur le suivi de trajectoire

### Suivi de trajectoire

- point de suivi  $P$ ;
- méthode simple de réduction d'erreur ;
- généralisable à une voiture.

## Conclusion sur le suivi de trajectoire

### Suivi de trajectoire

- point de suivi  $P$ ;
- méthode simple de réduction d'erreur ;
- généralisable à une voiture.

### Limitations

- $l_1 \neq 0$ ;
- contrôle de l'orientation ;
- prise en compte de la dynamique ;
- prise en compte du glissement.

## Conclusion sur le suivi de trajectoire

### Suivi de trajectoire

- point de suivi  $P$ ;
- méthode simple de réduction d'erreur ;
- généralisable à une voiture.

### Limitations

- $l_1 \neq 0$ ;
- contrôle de l'orientation ;
- prise en compte de la dynamique ;
- prise en compte du glissement.

### Suivi de chemin

- problème de la position de référence ;
- projection sur le repère de Frenet.

# 2

## Évitement d'obstacle



# Évitement d'obstacle

## Évitement d'obstacle

- utilisation des capteurs extéroceptifs ;
- génération de commandes permettant d'éviter les obstacles,
- tout en atteignant la cible.

## Approches

- champ de potentiel ;
- *vector field histogram* ;
- *dynamic window approach* ;
- *velocity obstacles*.

## *Vector field histogram*

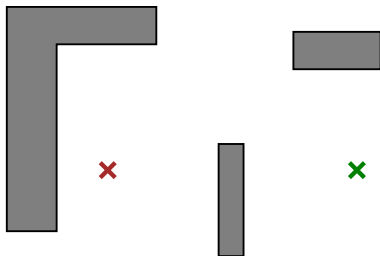
### *Vector field histogram*

- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.

## Vector field histogram

### Vector field histogram

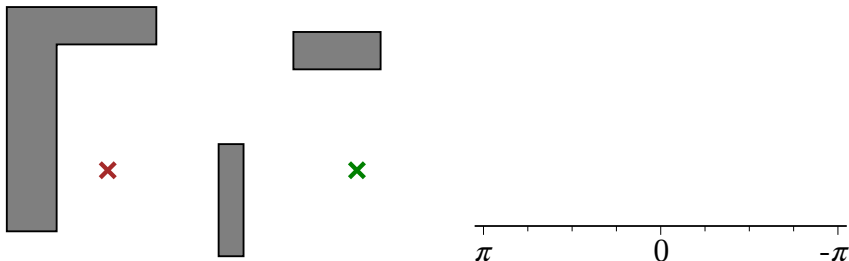
- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.



## Vector field histogram

### Vector field histogram

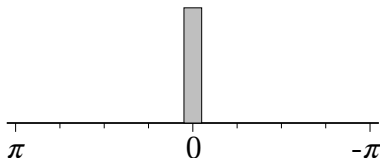
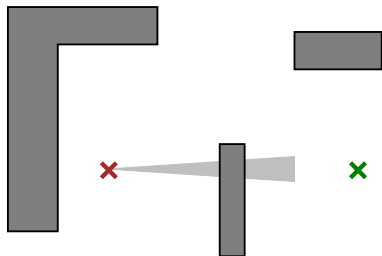
- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.



## Vector field histogram

### Vector field histogram

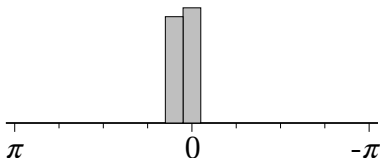
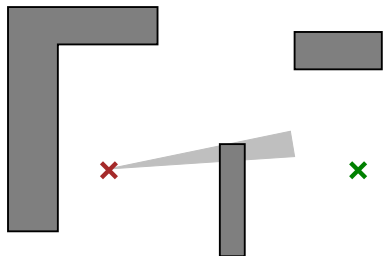
- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.



## Vector field histogram

### Vector field histogram

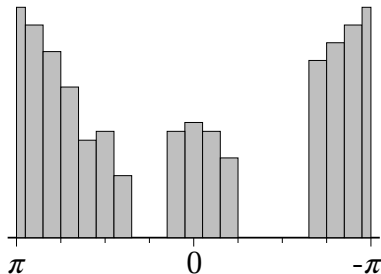
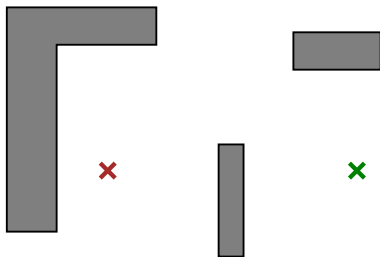
- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.



## Vector field histogram

### Vector field histogram

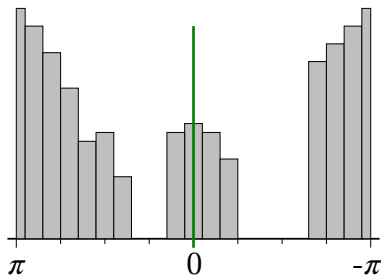
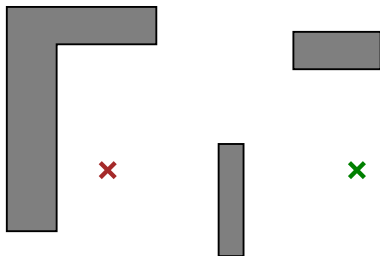
- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.



## Vector field histogram

### Vector field histogram

- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.

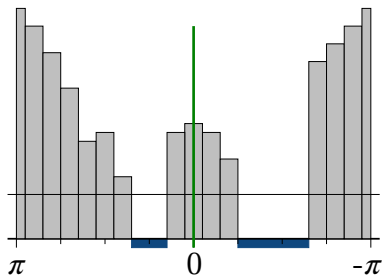
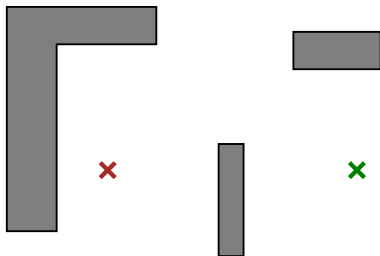




## Vector field histogram

### Vector field histogram

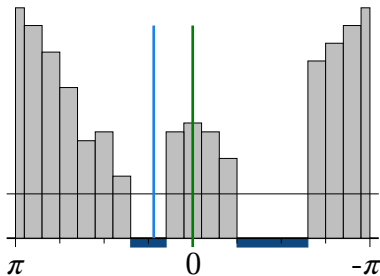
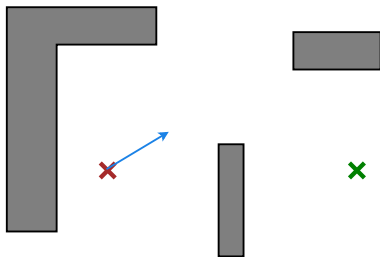
- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.



## Vector field histogram

### Vector field histogram

- histogramme de densité d'obstacle,
- suivant la direction ;
- identification de vallées ;
- choix de la vallée la plus proche.



## *Dynamic window approach*

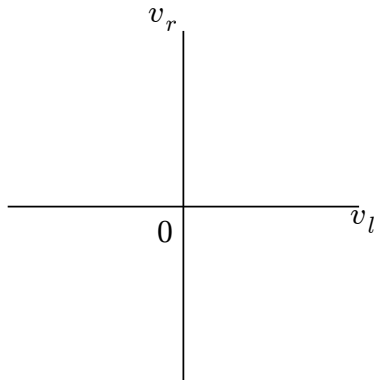
### *Dynamic window approach*

- espace des commandes ;
- identification des commandes menant à collision (horizon fini) ;
- identification des commandes réalisables au prochain pas de temps ;
- pondération entre :
  - commande désirée,
  - distance aux obstacles,
  - vitesses élevées.

## Dynamic window approach

### Dynamic window approach

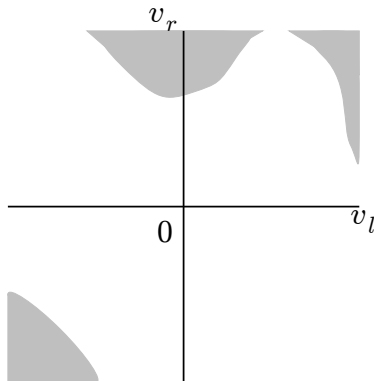
- espace des commandes ;
- identification des commandes menant à collision (horizon fini) ;
- identification des commandes réalisables au prochain pas de temps ;
- pondération entre :
  - commande désirée,
  - distance aux obstacles,
  - vitesses élevées.



## Dynamic window approach

### Dynamic window approach

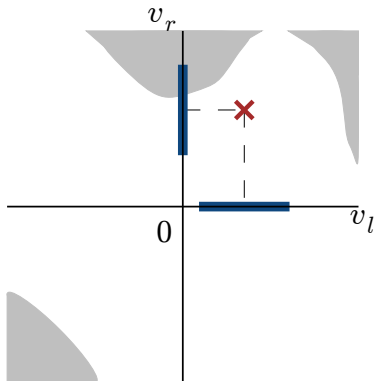
- espace des commandes ;
- identification des commandes menant à collision (horizon fini) ;
- identification des commandes réalisables au prochain pas de temps ;
- pondération entre :
  - commande désirée,
  - distance aux obstacles,
  - vitesses élevées.



## Dynamic window approach

### Dynamic window approach

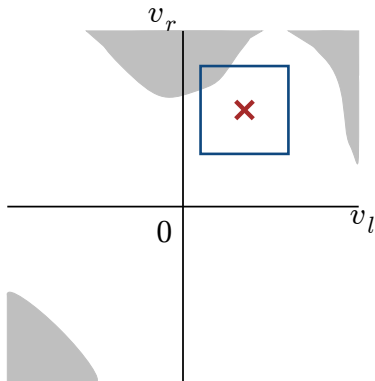
- espace des commandes ;
- identification des commandes menant à collision (horizon fini) ;
- identification des commandes réalisables au prochain pas de temps ;
- pondération entre :
  - commande désirée,
  - distance aux obstacles,
  - vitesses élevées.



## Dynamic window approach

### Dynamic window approach

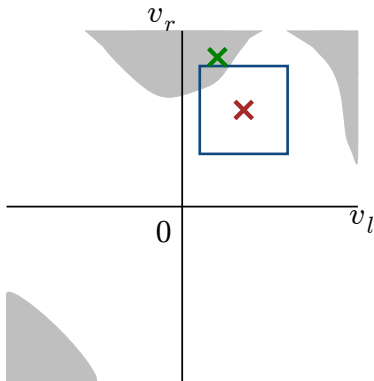
- espace des commandes ;
- identification des commandes menant à collision (horizon fini) ;
- identification des commandes réalisables au prochain pas de temps ;
- pondération entre :
  - commande désirée,
  - distance aux obstacles,
  - vitesses élevées.



## Dynamic window approach

### Dynamic window approach

- espace des commandes ;
- identification des commandes menant à collision (horizon fini) ;
- identification des commandes réalisables au prochain pas de temps ;
- pondération entre :
  - commande désirée,
  - distance aux obstacles,
  - vitesses élevées.

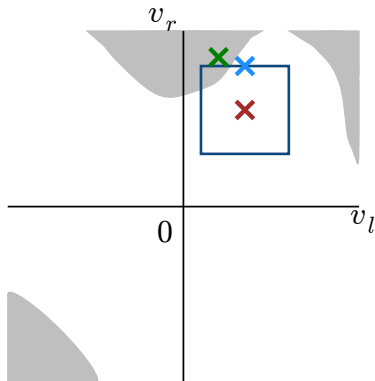




## Dynamic window approach

### Dynamic window approach

- espace des commandes ;
- identification des commandes menant à collision (horizon fini) ;
- identification des commandes réalisables au prochain pas de temps ;
- pondération entre :
  - commande désirée,
  - distance aux obstacles,
  - vitesses élevées.



## *Velocity obstacles*

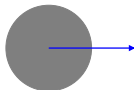
### *Velocity obstacles*

- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).

## *Velocity obstacles*

### *Velocity obstacles*

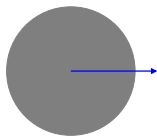
- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).



## *Velocity obstacles*

### *Velocity obstacles*

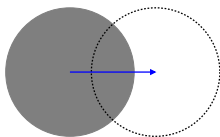
- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).



## Velocity obstacles

### Velocity obstacles

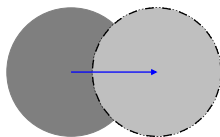
- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).



## *Velocity obstacles*

### *Velocity obstacles*

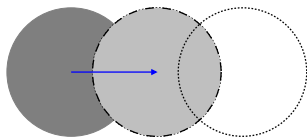
- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).



## Velocity obstacles

### Velocity obstacles

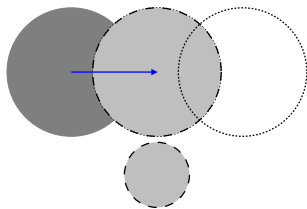
- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).



## Velocity obstacles

### Velocity obstacles

- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).

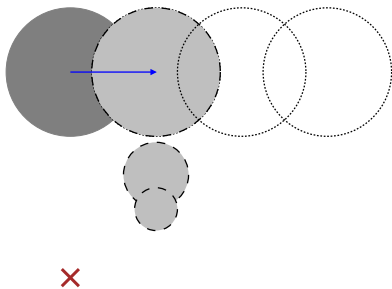




## Velocity obstacles

### Velocity obstacles

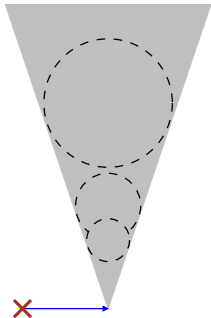
- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).



## *Velocity obstacles*

### *Velocity obstacles*

- évitement d'obstacles dynamiques ;
- vitesses connues ;
- planification dans l'espace des vitesses ;
- vitesses induisant une collision avec un obstacle (horizon infini).



## Conclusion sur l'évitement d'obstacle

### Évitement d'obstacle

- méthodes de modification locale de commande ;
- reconnaissance des commandes acceptables ;
- hors planification.

## Conclusion sur l'évitement d'obstacle

### Évitement d'obstacle

- méthodes de modification locale de commande ;
- reconnaissance des commandes acceptables ;
- hors planification.

### Limitations

- reconnaissance des obstacles ;
- estimation de leurs vitesses ;
- pas de garantie générale.

# 3

## Exploration

# Exploration

## Exploration

- choisir les actions d'un robot mobile,
- pour parcourir un environnement,
- en construisant sa carte ;
- optimisation de l'information.

# Exploration

## Exploration

- choisir les actions d'un robot mobile,
- pour parcourir un environnement,
- en construisant sa carte ;
- optimisation de l'information.

## *Active localization*

- localisation inconnue ;
- mouvements pour mieux se localiser ;
- SLAM + exploration.

# Exploration

## Exploration

- choisir les actions d'un robot mobile,
- pour parcourir un environnement,
- en construisant sa carte ;
- optimisation de l'information.

## *Active localization*

- localisation inconnue ;
- mouvements pour mieux se localiser ;
- SLAM + exploration.

## *Pursuit evasion problem*

- choisir les actions d'un robot mobile,
- pour trouver et suivre un autre objet mobile,
- dans un environnement connu ou non.



# Optimisation de l'information

## Quantité d'information

- utilisation de l'entropie :

$$H_p = \begin{cases} - \int p(x) \log p(x) dx \\ - \sum_x p(x) \log p(x) \end{cases}$$

- entropie : mesure d'incertitude ;
- maximiser l'information en minimisant l'entropie.

## Optimisation de l'information

### Quantité d'information

- utilisation de l'entropie :

$$H_p = \begin{cases} - \int p(x) \log p(x) dx \\ - \sum_x p(x) \log p(x) \end{cases}$$

- entropie : mesure d'incertitude ;
- maximiser l'information en minimisant l'entropie.

### Gain d'information

- comparaison entre l'information courante et l'information attendue :

$$I_p(u) = H_p - E[H_{p'} | u]$$

- observation prochaine inconnue ;
- résolution de POMDPs très coûteux.

# Heuristiques d'exploration

## Entropie

- corrélation entre l'entropie et le gain d'information dans une carte d'occupation ;
- méthode gloutonne : choisir la meilleure action immédiate.

# Heuristiques d'exploration

## Entropie

- corrélation entre l'entropie et le gain d'information dans une carte d'occupation ;
- méthode gloutonne : choisir la meilleure action immédiate.

## Incertitude dans l'inexploré

- plans à long termes invalides ;
- méthodes gloutonnes aux frontières.

## Heuristiques d'exploration

### Entropie

- corrélation entre l'entropie et le gain d'information dans une carte d'occupation ;
- méthode gloutonne : choisir la meilleure action immédiate.

### Incertitude dans l'inexploré

- plans à long termes invalides ;
- méthodes gloutonnes aux frontières.

### *Frontier-based exploration*

- lister les frontières connues ;
- aller explorer la plus proche.

# 4

## Conclusion

# Conclusion

## Navigation

- planification de mouvement ;
- adaptation au robot : suivi de trajectoire ;
- adaptation à l'environnement : évitement d'obstacles ;
- adaptation à notre connaissance : exploration.

# Conclusion

## Navigation

- planification de mouvement ;
- adaptation au robot : suivi de trajectoire ;
- adaptation à l'environnement : évitement d'obstacles ;
- adaptation à notre connaissance : exploration.

## Limites

- articulation entre la planification et l'exécution ;
- identification des obstacles ;
- exploration heuristique.



# Bibliographie

## Livres

- Siciliano *et al.*, *Springer Handbook of Robotics*, Springer 2016.
- Thrun *et al.*, *Probabilistic Robotics*, MIT Press, 2005.

## Vector Field Histogram

- Ulrich et Borenstein, *VFH+ : reliable obstacle avoidance for fast mobile robots*, RA, 1998.

## Velocity obstacles

- Fiorini et Shiller, *Motion planning in dynamic environments using velocity obstacles*, IJRR 1998.

## Exploration

- Holz *et al.*, *Evaluating the efficiency of frontier-based exploration*, ISR/Robotik, 2010.

Merci de votre attention.

Des questions ?