# RRT Path Planning using a Dynamic Vehicle Model

## Path Planning using a Dynamic Vehicle Model

Romain Pepy, Alain Lambert, Hugues Mounier

Information and Communication Technologies, 2006. ICTTA '06. 2nd

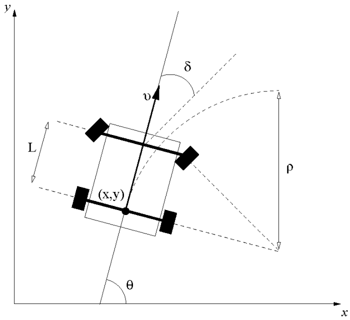
Ross Hatton, Siddharth Sanan

November 7, 2007

# Introduction

- Path planning for a robot with constraints
  - Simple: kinematic constraints
  - Harder: dynamic constraints
- RRT
- Runge-Kutta (RK4) integration

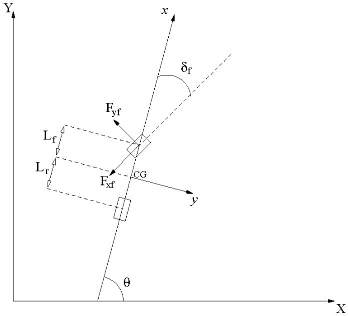
# Kinematic car




$$\begin{aligned}\dot{x} &= v\cos\theta \\ \dot{y} &= v\sin\theta \\ \dot{\theta} &= \frac{v}{L}\tan\delta\end{aligned}$$

# Dynamic car 1

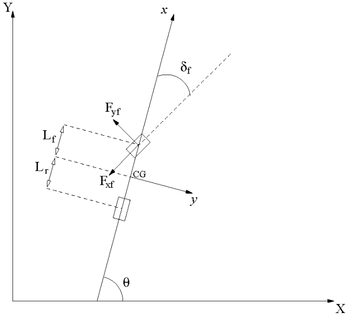
$$\mathbf{x} = (x_g, y_g, \theta, v_y, r)^T$$




$$\begin{aligned} m(\dot{v}_x - v_y r) &= -F_{xf}\cos\delta_f - F_{yf}\sin\delta_f - F_{xr} \\ m(\dot{v}_y + v_x r) &= F_{yf}\cos\delta_f - F_{xf}\sin\delta_f + F_{yr} \\ I_z \dot{r} &= L_f\left(F_{yf}\cos\delta_f - F_{xf}\sin\delta_f\right) - L_r F_{yr} \end{aligned}$$

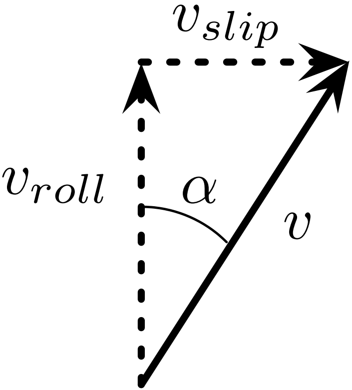
# Dynamic car 2

$$\begin{aligned}\dot{v}_y &= \frac{F_{yf}}{m}\cos\delta_f + \frac{F_{yr}}{m} - v_x r\\ \dot{r} &= \frac{L_f}{I_z}F_{yf}\cos\delta_f - \frac{L_r}{I_z}F_{yr}.\end{aligned}$$

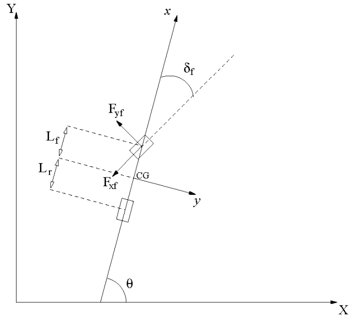


# Dynamic car 3 - Tire slip

$$
\begin{aligned}
F_{yf} &= -C_{\alpha f}\alpha_f & \alpha_f \\
F_{yr} &= -C_{\alpha r}\alpha_r & \alpha_r
\end{aligned}
$$



# Dynamic car 4



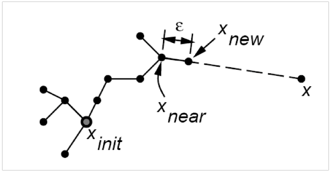

$$
\begin{aligned}
\dot{x}_g &= v_x \cos(\theta) - v_y \sin(\theta) \\
\dot{y}_g &= v_x \sin(\theta) + v_y \cos(\theta) \\
\dot{\theta} &= r
\end{aligned}
$$

$$
\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} A & C \\ B & D \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} E \\ F \end{bmatrix} \delta_f
$$

$$
\begin{aligned}
A &= -\frac{C_{\alpha f} \cos\delta_f + C_{\alpha r}}{m v_x}, \\
B &= \frac{-L_f C_{\alpha f} \cos\delta_f + L_r C_{\alpha r}}{m v_x} - v_x, \\
C &= \frac{-L_f C_{\alpha f} \cos\delta_f + L_r C_{\alpha r}}{I_z v_x}, \\
D &= -\frac{L_f^2 C_{\alpha f} \cos\delta_f + L_r^2 C_{\alpha r}}{I_z v_x}, \\
E &= \frac{C_{\alpha f} \cos\delta_f}{m}, \\
F &= \frac{L_f C_{\alpha f} \cos\delta_f}{I_z}.
\end{aligned}
$$

# Rapidly Exploring Random Trees

- A sampling based planner
- Tree $G$ is started by adding an initial point $x_{init}$
- A random configuration $x_{rand} \in X_{free}$ is chosen
- The nearest neighbor function finds $x_{near}$, the node of $G$ that offers the best approach to $x_{rand}$
- An input $u$ is chosen to drive the robot from $x_{near}$ towards $x_{rand}$
- Equations of motion integrated for a single time step to bring robot to $x_{new}$
- If $x_{new} \in X_{free}$, add it to $G$
- Path is found when $x_{new} \in X_{free} \cap X_{goal}$




**Algorithm 1** RRT [10]

**Function:** RRT($K \in \mathbb{N}$, $\mathbf{x}_{init} \in \mathbf{X}_{free}$, $\Delta t \in \mathbb{R}$)

$G$.init($\mathbf{x}_{init}$)
**for** $i = 0$ to $K$ **do**
$\mathbf{x}_{rand} \leftarrow$ random_config($\mathbf{X}_{free}$)
Extend($G$, $\mathbf{x}_{rand}$)
**end for**
**return** $G$

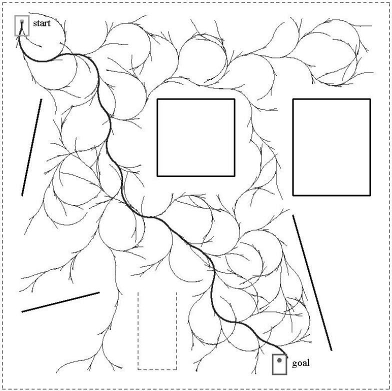
**Function:** Extend($G$, $\mathbf{x}_{rand}$)

$\mathbf{x}_{near} \leftarrow$ nearest_neighbor($G$, $\mathbf{x}_{rand}$)
$\mathbf{u} \leftarrow$ select_input($\mathbf{x}_{rand}$, $\mathbf{x}_{near}$)
$\mathbf{x}_{new} \leftarrow$ new_state($\mathbf{x}_{near}$, $\mathbf{u}$, $\Delta t$)
**if** collision_free_path($\mathbf{x}_{near}$,$\mathbf{x}_{new}$) **then**
$G$.add_node($\mathbf{x}_{new}$)
$G$.add_edge($\mathbf{x}_{near}$, $\mathbf{x}_{new}$, $\mathbf{u}$)
**end if**
**return** $G$

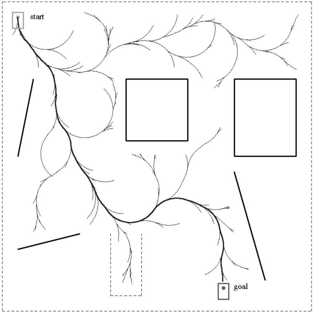
# RRTs with kinematic constraints

$$
\begin{aligned}
\dot{x} &= v \cos\theta \\
\dot{y} &= v \sin\theta \\
\dot{\theta} &= \frac{v}{L} \tan\delta
\end{aligned}
$$

- $\delta$ is the controlled variable
- Kinematic equations restrict allowable paths
- new_state function respects restriction
- Any path found by the planner will respect the nonholonomic constraints on the car



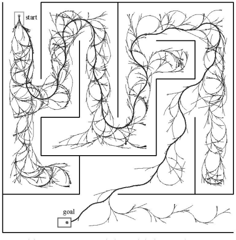
# RRT with dynamic constraints



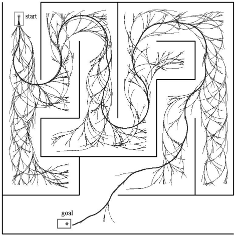

- Greater lower bound on turning radius
- Smoother paths
- Tighter turns at speed cause slip
- Fewer open paths in $X_{free}$

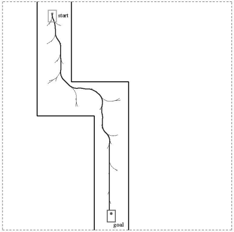
# Relative performance

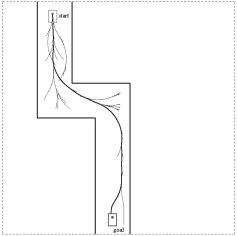
Kinematic




Dynamic



# Computation time comparison

- Approximately equal iteration time for both models.
- Slightly higher integration time on dynamic model to deal with extra degrees of freedom.
- Significant increase in *overall* computation time for dynamic model.

| | Kinematic | Dynamic |
|---|---|---|
| Time (s) | 4.4 | 11.8 |

- Not all iterations produce a node in $X_{free}$.
- Computation time lost each time a node is thrown out.
- Dynamic vehicle can't turn as fast, so more bad nodes.
- 30k nodes needed for kinematic model, 70k for dynamic model.

# Runge-Kutta integration 1

- Numerical integration
- Euler's method doesn't account for changes in force during time step
  - Error $O(\Delta t)$
- Runge-Kutta methods add extra terms
  - Weighted average of integrand at $t_n$, $t_n + \frac{\Delta t}{2}$ and $t_n + \Delta t$
  - More computational cost per step than Euler
  - Error $O(\Delta t^4)$

# Runge-Kutta integration 2

$$\mathbf{x}_{n+1} \approx \mathbf{x}_n + \frac{\Delta t}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

$$\begin{aligned}
k_1 &= f(\mathbf{x}_n, \mathbf{u}_n)_{t_n}, \\
k_2 &= f\left(\mathbf{x}_n + \frac{k_1}{2}, \mathbf{u}_n\right)_{t_n + \frac{\Delta t}{2}}, \\
k_3 &= f\left(\mathbf{x}_n + \frac{k_2}{2}, \mathbf{u}_n\right)_{t_n + \frac{\Delta t}{2}}, \\
k_4 &= f\left(\mathbf{x}_n + k_3, \mathbf{u}_n\right)_{t_n + \Delta t},
\end{aligned}$$

# Conclusions

- Path planning for a robot with constraints
  - Simple: kinematic constraints
  - Harder: dynamic constraints
- RRT
- Runge-Kutta (RK4) integration

# Robotic Motion Planning: Probabilistic Primer

Robotics Institute 16-735

http://www.cs.cmu.edu/~motion


Howie Choset

http://www.cs.cmu.edu.cmu.edu/~choset

# Overview

- Ross and Sidd’s talk
- Reminder of pending HW Assignment
- Next Wednesday’s mini-project report
- Misc. discussion

# Experiments and outcomes

- Experiment: Flip a Coin
- Outcome: Heads or Tails

- Experiment: Person’s temperature in class now
- Outcome: a scalar

- Event Subset of possible outcomes $E \subset S$

# Probability

- Pr(*E*): Probability of an event *E* occurring when an experiment is conducted
- Pr maps *S* to unit interval

$$\Pr(\text{heads}) = 0.5, \Pr(\text{tails}) = 0.5$$

$$\Pr(\text{heads} \cup \text{tails}) = 1$$

1. $0 \leq \Pr(E) \leq 1$ for all $E \subset \mathcal{S}$.

2. $\Pr(\mathcal{S}) = 1$.

3. $\sum_i \Pr(E_i) = Pr(E_1 \cup E_2 \cup \ldots)$ for any countable disjoint collection of sets $E_1, E_2, \ldots$. This property is known as *sigma additivity*. In particular, we have $\sum_{i=1}^{n} \Pr(E_i) = Pr(E_1 \cup E_2 \cup \ldots \cup E_n)$.

4. $\Pr(\emptyset) = 0$.

5. $\Pr(E^c) = 1 - \Pr(E)$, where $E^c$ denotes the complement of $E$ in $\mathcal{S}$.

6. $\Pr(E_1 \cup E_2) = \Pr(E_1) + \Pr(E_2) - \Pr(E_1 \cap E_2)$.

# Dependence

- Independent: $\Pr(E_1 \cap E_2) = \Pr(E_1)\Pr(E_2)$

- Conditional Probability:

  – If $E_1$ and $E_2$ are independent, then $\Pr(E_1|E_2) = \Pr(\mathrm{E}_1)$

  – Bayes Rule

$$\Pr(E_1|E_2) = \frac{\Pr(E_2|E_1)\Pr(E_1)}{\Pr(E_2)}.$$

# More Bayes Rule

$$p(a|b) = \frac{p(b|a)\ p(a)}{p(b)}$$

$$p(a|b,c) = \frac{p(b|a,c)\ p(a|c)}{p(b|c)}$$

# Total Probability

Discrete

$$p(a) = \sum_i p(a \wedge b_i)$$
$$= \sum_i p(a \mid b_i)\, p(b_i)$$

Continuous

$$p(a) = \int p(a \mid b)\, p(b)\, db$$

it follows that:

$$p(a \mid b) = \int p(a \mid b, c)\, p(c \mid b)\, dc$$

# Random Variable

- A mapping from events to a real number $X : \mathcal{S} \rightarrow \mathbb{R}$

- Examples
  - Discrete: heads or tails,  number of heads for repeated flips
  - Continuous: temperature

- Random Vector: $X : \mathcal{S} \rightarrow \mathbb{R}^n$

# Distribution

- Any statement one can make about a random variable

- Cumulative Distribution Function (CDF): $F_X(a) = \Pr(X \leq a)$

- Probability ______ Function (P_F):

  - Discrete: Mass (PMF) $f_X(a) = \Pr(X = a)$

  - Continuous: Density (PDF) $\Pr(a \leq X \leq b) = \int_{x=a}^{b} f_X(x)dx$

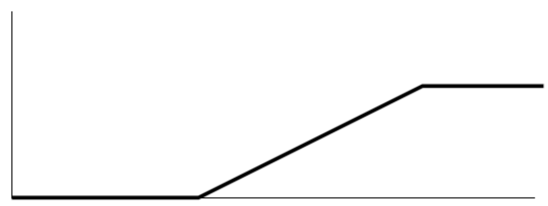
Note that $Pr((X = a)) = \int_{x=a}^{a} f_X(x)dx$ = 0

# Uniform Distribution

CDF

$$U(x;a,b) = \begin{cases} 0 & x < a \\ \frac{x-a}{b-a} & a \leq x \leq b \\ 1 & x \geq b \end{cases}$$

$$Pr(a' \leq x \leq b') = U(b';a,b) - U(a';a,b)$$



PDF

$$u(x;a,b) = \begin{cases} 0 & x < a \\ \frac{1}{b-a} & a \leq x \leq b \\ 0 & x \geq b. \end{cases}$$

$$Pr(a' \leq x \leq b') = \int_{x=a'}^{b'} u(x;a,b)dx$$

# Expected Value

- PMF

$$E(X) = \sum_i x_i f_X(x_i)$$

PDF

$$E(X) = \int_{x \in \mathbb{R}^n} x f_X(x) dx$$

$$E(aX + bY) = aE(X) + bE(Y)$$

$$E(X) \text{ with } \bar{X}$$

# Variance and Co-variance

- Variance: $\sigma^2 = E((X - \bar{X})^2)$

  Or $\sigma_i^2$ for $X_i$

- Co-Variance $\sigma_{ij} = E((X_i - \bar{X}_i)(X_j - \bar{X}_j))$

  $\sigma_{ij} = 0$, $X_i$ and $X_j$ are independent

- Co-Variance Matrix $P_X = E\left((X - \bar{X})(X - \bar{X})^T\right)$

  $\sigma_i^2$ Diagonal terms

  $\sigma_{ij}$ Off-Diagonal Terms

# Gaussians

$$f_X(x) = \frac{1}{\sqrt{(2\pi)^n |P_X|}} e^{-\frac{1}{2}(x-\bar{X})^T P_X^{-1} (x-\bar{X})}$$

$P_X \in \mathbb{R}^{n \times n}$ $\qquad$ $\bar{X} \in \mathbb{R}^n$ is the mean vector