Planning Under Differential Constraints

Velocity Constraints

Ограничения в неявной форме:

$$g(q,\dot{q})=0 \qquad (1)$$

Примеры ограничений для скорости:

$$\dot{x}>0$$

$$a\dot{x}+b\dot{y}+c=0$$

Ограничение для максимальной скорости:

$$\dot{x}^2+\dot{y}^2\leq 1$$

Движение точки на плоскости

Положение задается координатами

$$q = q(x, y) \in R^2$$

Скорость является элементом **tangent space *T_q*** $(\dot{x}, \dot{y}) \in T_q(R^2)$

Для каждой точки с координатами q(x,y) задается вектор действия:

$$u \in U(q)$$

Ограничение в параметрической форме:

$$\begin{aligned} \dot{x} &= f_1(x, y, u) \\ \dot{y} &= f_2(x, y, u) \end{aligned} \qquad (2)$$

В сокращенной форме уравнение (2) имеет вид:

$$\dot{q} = f(q, u) \qquad (3)$$

Ограничения в параметрической форме в N-мерном случае

Положение задается координатами

$$q = q(q_1, q_2, ..., q_N) \in C_{space}$$

Скорость является элементом **tangent space *T_q***

$$(\dot{q}_1, \dot{q}_2, ..., \dot{q}_N) \in T_q$$

Вектор действия:

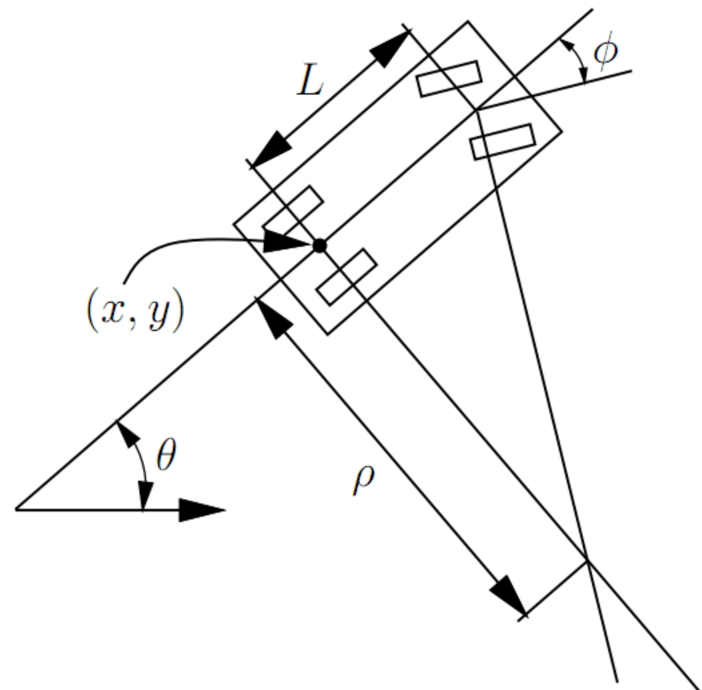
$$u \in U(q)$$

Ограничение в параметрической форме:

$$\dot{q} = f(q, u) \tag{4}$$

Simple car

$$q = q(x, y, \theta) \qquad \dim(C_{space}) = 3$$



Transition equation:

$$\dot{x} = u_s \cos(\theta)$$

$$\dot{y} = u_s \sin(\theta)$$

$$\dot{\theta} = \frac{u_s}{L} tg(u_\varphi)$$

$$u = (u_s, u_\varphi)$$

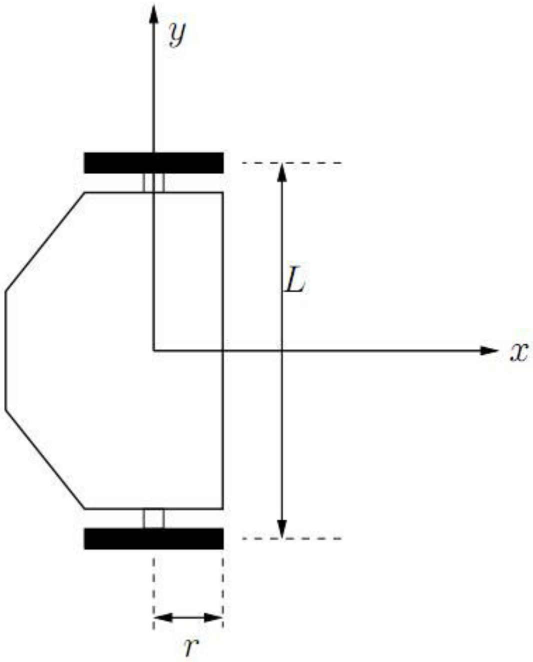
u_s – *скорость машины*

u_φ – *угол* φ

$$|u_\varphi| < \frac{\pi}{2}, \quad |u_s| < v_{\max}$$

Kinematics for Wheeled Systems

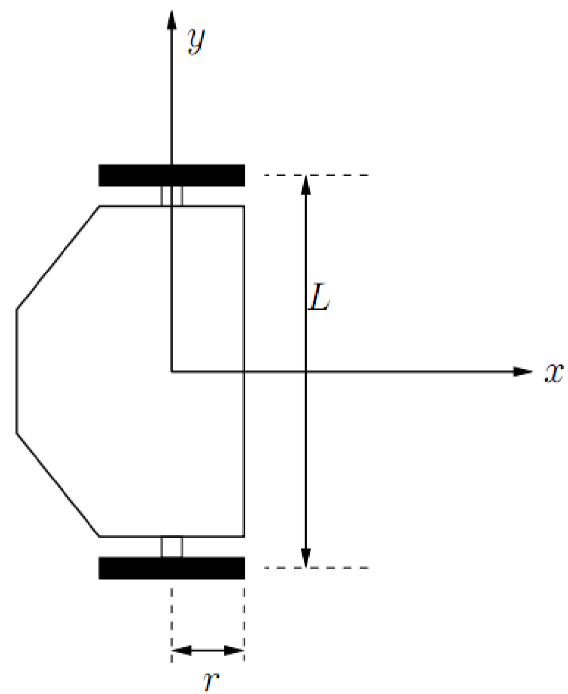
(a)



(b)

Kinematics for Wheeled Systems

$$q = q(x, y, \theta) \qquad \dim(C_{space}) = 3$$



$$\dot{x} = \frac{r}{2}(u_l + u_r)\cos(\theta)$$

$$\dot{y} = \frac{r}{2}(u_l + u_r)\sin(\theta)$$

$$\dot{\theta} = \frac{r}{L}(u_r - u_l)$$

$u = (u_l, u_r)$ *задает угловую скорость каждого колеса*

Phase Space Representation

<u>**Фазовое пространство**</u> **(Phase Space: $\mathbf{X}$)** определяет множество всех состояний системы в фиксированный момент времени.
Элемент пространства задается обобщёнными координатами q_i и обобщёнными импульсами p_i (i = 1, 2,..., N).

Ограничение для ускорения:

$$\ddot{q} = f(\dot{q}, q, u) \tag{5}$$

Введем обозначения:

$$x_1 = q$$

$$x_2 = \dot{q}$$

Тогда уравнение (5) запишется в виде:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(x_2, x_1, u) \end{aligned} \tag{6}$$

Движение частицы

2-й закон Ньютона:

$$\ddot{q} = \frac{f}{m}$$

В одномерном случае:

$$X = R^2$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{u}{m}$$

$$u = f$$

$$U = [-f_{\max}, f_{\max}]$$

Движение частицы

В двумерном случае:

$$X = R^4$$

$$\begin{cases} \dot{x}_1 = x_3 \\ \dot{x}_2 = x_4 \\ \dot{x}_3 = \dfrac{u_1}{m} \\ \dot{x}_4 = \dfrac{u_2}{m} \end{cases}$$

$u_1, u_2 - \textit{компоненты вектора силы}$

$U = \{u \in R^2 \mid \|u\| \le f_{\max}\}$

Lunar Lander

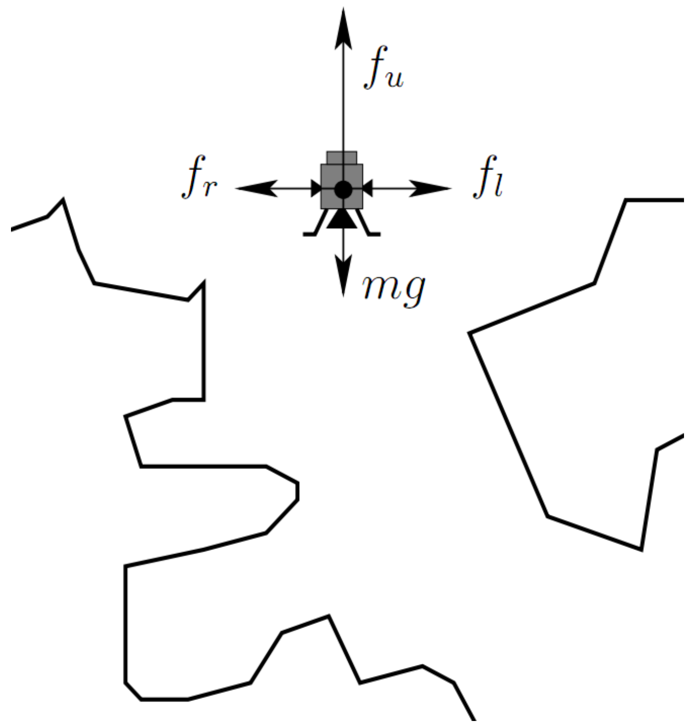
$$\dim(C_{space}) = 2,\ X = R^4$$

$$\dot{x}_1 = x_3$$

$$\dot{x}_2 = x_4$$

$$\dot{x}_3 = \frac{f_s}{m}(u_l f_l - u_r f_r)$$

$$\dot{x}_4 = \frac{u_u f_u}{m} - g$$



Each action vector is of the form (u_l, u_r, u_u),
in which each component is 0 *or* 1.

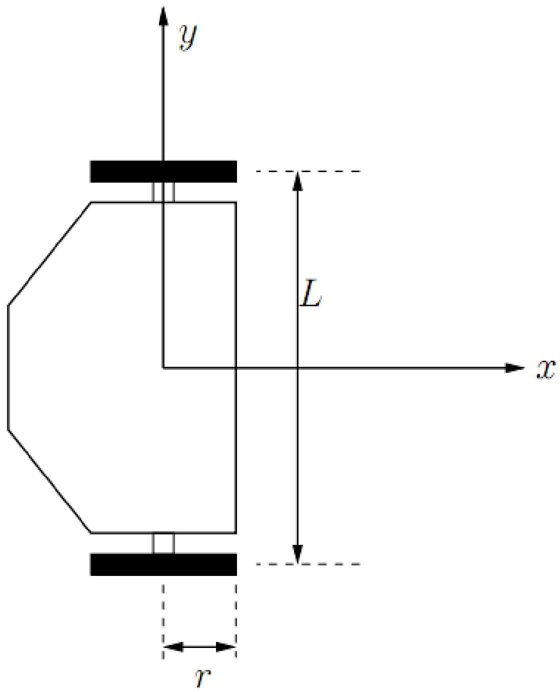
Kinematics for Wheeled Systems

$$\dim(C_{space}) = 3, \ \dim(X) = 6$$

$$\dot{x} = \frac{r}{2}(u_l + u_r)\cos(\theta)$$

$$\dot{y} = \frac{r}{2}(u_l + u_r)\sin(\theta)$$

$$\dot{\theta} = \frac{r}{L}(u_r - u_l)$$



Функции $u = (u_l, u_r)$ *задают угловые скорости колес и должны быть непрерывными.*

Second-order differential drive model

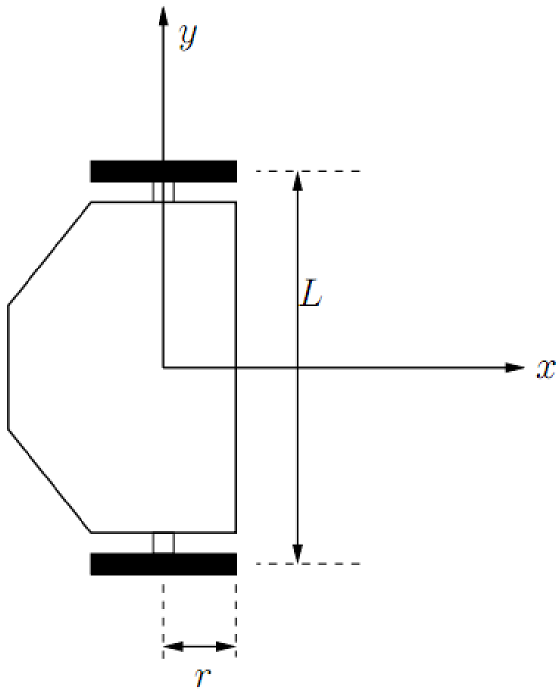
$$\dot{x} = \frac{r}{2}(\omega_l + \omega_r)\cos(\theta)$$

$$\dot{y} = \frac{r}{2}(\omega_l + \omega_r)\sin(\theta)$$

$$\dot{\theta} = \frac{r}{L}(\omega_r - \omega_l)$$

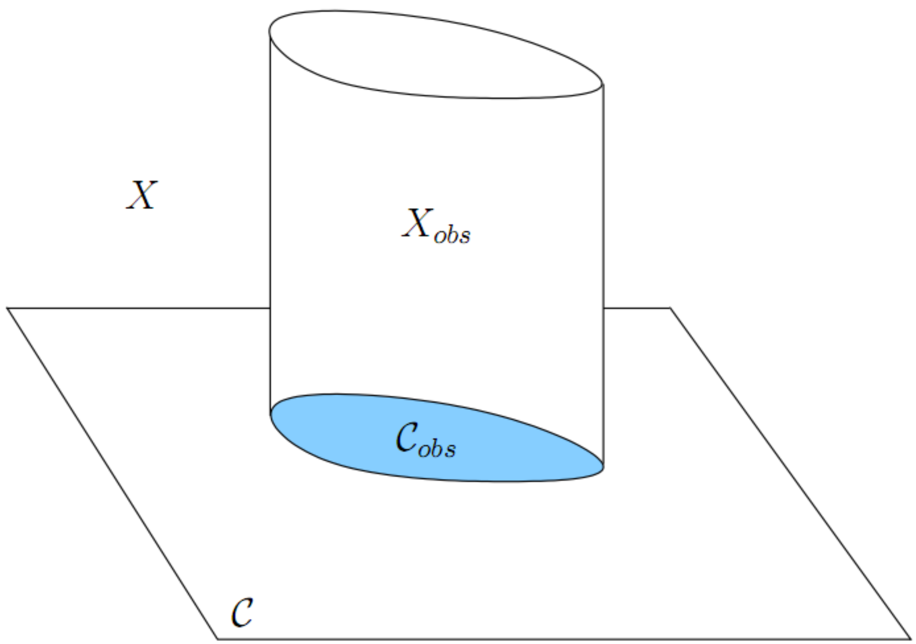
$$\dot{\omega}_l = u_l$$

$$\dot{\omega}_r = u_r$$



Функции $u = (u_l, u_r)$ *задают угловое ускорение.*

Obstacles in the Phase Space



$$X_{obs} = \{ x \in X \mid \kappa(x) \in C_{obs} \}$$

$$X_{free} = X \setminus X_{obs}$$

Additional constraints on phase variables

Для скорости:

$$\|\dot{q}\| \leq \dot{q}_{\max}$$

Для ускорения:

$$\|\ddot{q}\| \leq \ddot{q}_{\max}$$

Action trajectory

Ограничения заданы в параметрическом виде:

$$\dot{x} = f(x, u) \tag{7}$$

в фазовом пространстве X.

Путь U – пространство действий (action space).

Алгоритм строит *action trajectory* $\tilde{u}$, которая задается

функией вида $\tilde{u}:\ [0, \infty) \to U$

Связь между траекторией движения робота и *action trajectory* задается формулой:

$$x(t) = x(0) + \int_0^t f(x(t'), u(t'))dt' \tag{8}$$

Problem Formulation

1. world W, robot A, obstacle region O,
   configuration space C.

2. An unbounded time interval $T =[0,\infty)$.

3. A manifold X, called the state space. It may be a phase space derived from C if dynamics is considered.

4. An obstacle region X_{obs} is defined for the state space.

5. Множество возможных действий U для каждого элемента из X.

Problem Formulation

6. A system is specified using a state transition equation

$$\dot{q} = f(q,u)$$

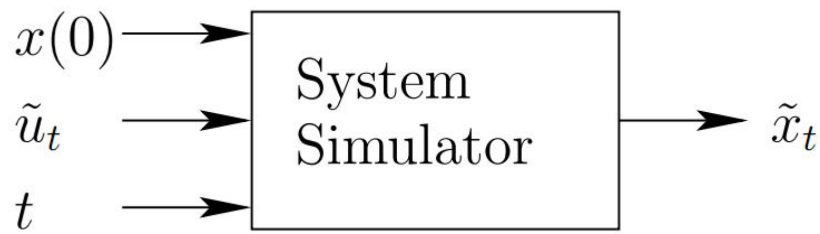
7. A state $x_I \in X_{free}$

8. A set $X_G \subset X_{free}$

9. A complete algorithm must compute an **action trajectory**, for which the **state trajectory**, satisfies:
   1) $x(0) = x_I$
   2) there exists some $t > 0$ for which $u(t) = u_T$ and $x(t) \in X_G$.

System Simulator

$$\dot{q} = f(q,u)$$

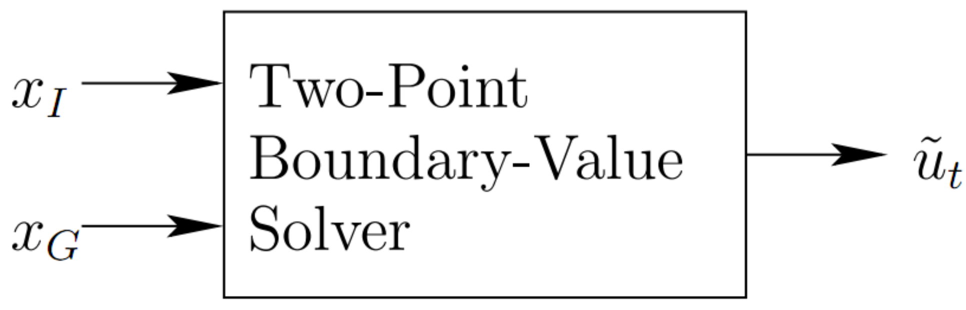


$$x(t) = x(0) + \int_0^t f(x(t'),u(t'))dt'$$

Euler integration method:

$$x(\Delta t) = x(0) + f(x(0),u(0))\Delta t$$

Local Planning



Efficiently connects x_I to x_G in the absence of obstacles.

$$\dot{q} = f(q, u)$$

Sampling-Based Motion Planning

1. **Initialization:** Let G(V, E) represent an undirected search graph, for which the vertex set V contains a vertex x_I, and the edge set E is empty.

2. **Swath-point Selection Method (SSM):** Choose a vertex x_{cur} for expansion.

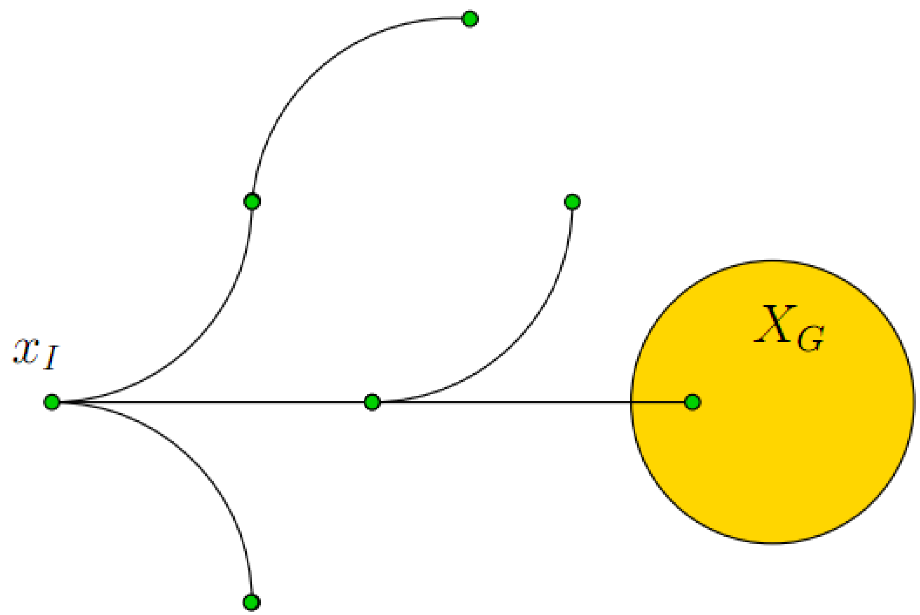
3. **Local Planning Method :** Generate a motion primitive such that $u(0)= x_{cur}$ and $u(t_F)= x_r$ for some $x_r \in X_{free}$, which may or may not be a vertex in G. Using the system simulator, a collision detection algorithm, and by testing the phase constraints, it must be verified to be violation-free. If this step fails, then go to Step 2.

4. **Insert an Edge in the Graph**

5. **Check for a Solution**

6. **Return to Step2:** Iterate unless a solution has been found or some termination condition is satisfied.

Sampling-Based Motion Planning



The situation, in which forward, unidirectional search is used to enter a large goal region.

Rapidly exploring dense tree(RDT)

SIMPLE_RDT_WITH_DIFFERENTIAL_CONSTRAINTS(x_0)

$\mathcal{G}$.init(x_0);

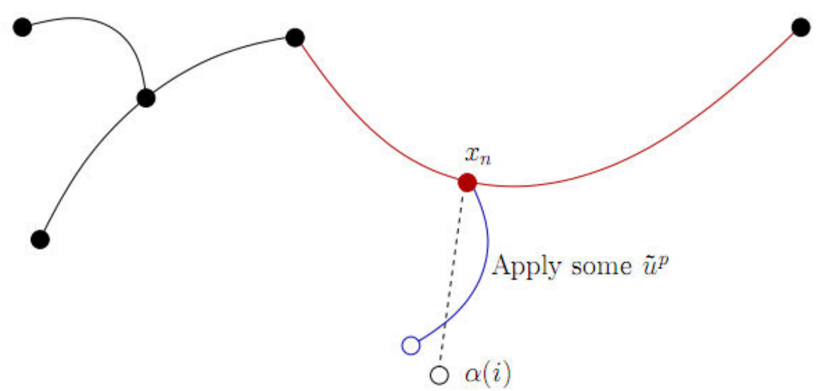
for $i = 1$ **to** k **do**

  $x_n \leftarrow$ NEAREST$(S(\mathcal{G}), \alpha(i))$;

  $(\tilde{u}^p, x_r) \leftarrow$ LOCAL_PLANNER$(x_n, \alpha(i))$;

  $\mathcal{G}$.add_vertex(x_r);

  $\mathcal{G}$.add_edge($\tilde{u}^p$);



Nearest point S lies in the state trajectory segment associated to an edge