The Dynamic Window Approach to Collision Avoidance

⇒ This approach, designed for mobile robots equipped with synchro-drives.

⇒ It is derived directly from the motion dynamics of the robot.

⇒ " Reactive avoidance of collisions with obstacles "

Admissible Velocities → { Velocity at which robot can stop safely }

⇒ Acceleration of motor is also limited which further ~~restrict~~ impose restriction on the velocities. → Reachable Velocities

⇒ These velocities form the dynamic window which is centered around the current velocities of the robot in the velocity space.

⇒ Among the admissible velocities within the dynamic window the combination of translational and rotational velocity is chosen by maximizing an Objective function.

⇒ The objective function includes :
- → measure of progress toward goal location
- → Forward velocity of the robot.
- → Distance to the next obstacle on the trajectory.

★ Related Work

⇒ The Collision avoidance approache for mobile robots can roughly be divided into two categories:

- Global
  {road-map, cell decomposition & Potential field method}
- Local
  {They cannot produce optimal solution}
  → low computational complexity.

★ Motion Equations for a Synchro Drive Robot

⇒ Robot's translational and rotational velocity can be controlled independently.
{With limited torque}

⇒ To make the equation more practical, we derive an approximation that models velocity as a piece-wise constant function in time.

- So robot trajectories consist of sequence of finitely many segments of circle.

⇒ We also derive upper bound for the approximation error.

Scanned by CamScanner

* General Motion Equation

⇒ Let $x(t)$ and $y(t)$ denote the robots' coordinate at time t in some global coordinate system and let the robot's orientation be described by $\theta(t)$.

⇒ The triplet $\langle x, y, \theta \rangle$ describe the Kinematic Configuration of the robot.

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} V(t) \cdot \cos \theta(t)\, dt \quad \text{—— ①}$$

$$y(t_n) = y(t_0) + \int_{t_0}^{t_n} V(t) \cdot \sin \theta(t)\, dt \quad \text{—— ②}$$

$$V(t) = V(t_0) + \int_{t_0}^{t} \dot{V}(t)\, dt \quad \text{—— ③}$$

$$\theta(t) = \theta(t_0) + \int_{t_0}^{t} \omega(t)\, dt \quad \text{—— ④}$$

$$\omega(t) = \omega(t_0) + \int_{t_0}^{t} \dot{\omega}(t)\, dt \quad \text{—— ⑤}$$

⇒ The trajectory of the robot depends exclusively on its initial dynamic configuration at time t_0, and the accelerations,

(It is assumed to be controllable)

⇒ Digital hardware impose constraints as to when one can set the motor currents.

⇒ Between two arbitrary point in time t_0 and t_m, the robot can only be controlled by finitely many acceleration commands.

⇒ Then, the acceleration command $\dot{v}_i$ and $\dot{\omega}_i$ are kept constant in the time interval $[t_i, t_{i+1}]$

Let $\Delta_t^i = t - t_i$

★ Approximate Motion Functions

$$x(t_n) = x(t_0) + \int_{t_0}^{t_n} v(t) \cdot \cos\theta(t)\, dt$$

&

$$v(t) = v(t_0) + \int_{t_0}^{t} \dot{v}(t)\, dt$$

$$\theta(t) = \theta(t_0) + \int_{t_0}^{t} \omega(t)\, dt$$

$$\omega(t) = \omega_0(t) + \int_{t_0}^{t} \dot{\omega}(t)\, dt$$

$$x(t_n) = x(t_{n-1}) + \int_{t_{n-1}}^{t_n} \left(v(t_{n-1}) + \int_{t_{n-1}}^{t} \dot{v}(t)\, dt \right) *$$

$$\cos\left(\theta(t_{n-1}) + \int_{t_{n-1}}^{t} \omega_0(t_{n-1}) + \int_{t_{n-1}}^{t'} \dot{\omega}(t)\, d(t') \right) dt$$

$$x(t_n) = x(t_{n-1}) + \int_{t_{n-1}}^{t_n} \Big(V(t_{n-1}) + \dot{V}(t_{n-1})(t - t_{n-1}) \Big)$$

$$* \cos\Big(\theta(t_{n-1}) + \omega(t_{n-1})(t - t_{n-1}) + \frac{1}{2}\dot{\omega}(t_{n-1})(t - t_{n-1})^2\Big) dt$$

$$\boxed{x(t_n) = x(t_0) + \sum_{i=0}^{n-1} \int_{t_i}^{t_{i+1}} \left(V(t_i) + \dot{V}_i \Delta_t^i \right) * \cos\left(\theta(t_i) + \omega(t_i)\cdot\Delta_t^i + \frac{1}{2}\dot{\omega}_i (\Delta_t^i)^2\right) dt}$$

{Obtained after constant acceleration approximation}

$$\boxed{\begin{aligned} x(t_n) &= x(t_0) + \sum_{i=0}^{n-1} \left(F_x^i(t_{i+1}) \right) \\ \text{where } F_x^i(t) &= \begin{cases} \dfrac{V_i}{\omega_i}\Big(\sin\theta(t_i) - \sin\big(\theta(t_i) + \omega_i\cdot(t - t_i)\big)\Big) & \forall\, \omega_i \neq 0 \\ V_i \cos(\theta(t_i)) \cdot t & \forall\, \omega_i = 0 \end{cases} \end{aligned}} \quad (1)$$

{By approximating the robots velocities within a time interval $[t_i, t_{i+1}]$ by a constant value}

⇒ The corresponding equations for the y-coordinate are :

$$\boxed{\begin{aligned} y(t_n) &= y(t_0) + \sum_{i=0}^{n-1} \left(F_y^i(t_{i+1}) \right) \\ F_y^i(t) &= \begin{cases} \frac{-V_i}{\omega_i} \left(\cos\theta(t_i) - \cos\left(\theta(t_i) + \omega_i \cdot (t - t_i)\right)\right) & \forall\, \omega_i \neq 0 \\ V_i \sin\left(\theta(t_i)\right) * t & \forall\, \omega_i = 0 \end{cases} \end{aligned}}$$

(2)

⇒ Equation (1) & (2) only depends on velocities.

↳ When controlling the robot however, one is not free to set arbitrary velocities, since dynamic constraints of the robot impose bounds on the maximum deviation of velocity value in subsequent interval.

* Upper bound on approximation Error

⇒ Consider the error E_x^i and E_y^i for the x and y coordinate respectively, within the time interval $[t_i ; t_{i+1}]$

⇒ Let $\Delta t_i = t_{i+1} - t_i$

⇒ The deviation in the direction of any of the two axis is maximum if the robot moves on a straight trajectory parallel to that axis.

Scanned by CamScanner

⇒ Since in each time interval we approximate $V(t)$ by an arbitrary velocity $V_i \in [V(t_i), V(t_{i+1})]$

↳ An upper bound of the error E_x^i and E_y^i for $(i+1)^{th}$ time interval is governed by

$$\boxed{E_x^i , E_y^i \leq |V(t_{i+1}) - V(t_i)| \cdot \Delta t_i}$$

★ Dynamic Window Approach

⇒ In the dynamic window approach, the search for commands controlling the robot is carried out directly in the space of velocities.

⇒ The dynamics of the robot is incorporated into the method by reducing the search space to those velocities which are reachable under dynamic constraints.

↳ In addition to this restriction only velocities are considered which are safe with respect to the obstacles.

⇒ This Algo runs in two step :-

1. Search Space

→ The search space of the possible velocities is reduced in three space.

ⓐ Circular trajectories

⇒ dynamic window approach considers only circular trajectories uniquely determined by (V, ω) pair.

⇒ This results in 2D velocity search space.

Scanned by CamScanner

ⓑ <u>Admissible Velocities</u>

A pair (V, ω) is considered admissible, if the robot is able to stop before it reaches the closest obstacle on the corresponding curvature.

ⓒ <u>Dynamic Window</u>

⇒ Restrict admissible velocities to those that can be reached within a short time interval given the limited acceleration of the robot.

2. <u>Optimization</u>

→ The objective function:

$$G(V,\omega) = \sigma\left(\alpha \cdot \text{heading}(V,\omega) + \beta \cdot \text{dist}(V,\omega) + \gamma\, \text{Vel}(V,\omega)\right)$$

is maximized.

→ With respect to current position and orientation of the robot this function trades off the following aspects:

ⓐ <u>Target heading</u>

Measure of progress toward the goal location. It is maximal if the robot moves directly towards the target.

ⓑ <u>Clearance</u>

⇒ dist is the distance to the closest obstacle on the trajectory.

⇒ The smaller the distance to an obstacle the higher is the robot's desire to move around it.

(c) Velocity

→ Vel is forward velocity of the robot and supports fast movements.

⇒ The function σ smooths the weighted sum of the three components and results in more side-clearance from obstacles.

★ Circular Trajectories

⇒ To generate ~~Ea~~ trajectory to a given goal point for the next n time intervals the robot has to determine velocities (V_i, ω_i) one for each of the n intervals between t_0 to t_n.

⇒ Dynamic window approach considers exclusively the first time interval and assumes that the velocities in the remaining n-1 time intervals are constant.

★ Admissible Velocities

⇒ Let $\dot{V}_b$ and $\dot{\omega}_b$ be the accelerations for breakage.

⇒ Then the set V_a of admissible velocities is defined as:

$$V_a = \left\{ V, \omega \;\middle|\; V \le \sqrt{2 \cdot dist(V,\omega) \cdot \dot{V}_b} \wedge \omega \le \sqrt{2\, dist(V,\omega) \cdot \dot{\omega}_b} \right\}$$

★ Dynamic Window

⇒ Let t be the time interval during which the accelerations $\dot{V}$ and $\dot{\omega}$ will be applied and let (V_a, ω_a) be the actual velocity.

⇒ Then the dynamic window V_d is defined as:

$$V_d = \{(v,\omega) \mid v \in [v_a - \dot{v}t, v_a + \dot{v}t] \wedge \omega \in [\omega_a - \dot{\omega}t, \omega_a + \dot{\omega}t]\}$$

★ Resulting Search Space

$$\underbrace{V_r}_{\text{Resultant Search Space}} = \underbrace{V_s}_{\text{Space of all Possible velocities}} \cap \underbrace{V_a}_{\text{admissible Velocities}} \cap \underbrace{V_d}_{\text{dynamic window}}$$

★ Maximizing the Objective function

⇒ After having determined the resulting Search space V_r, a velocity is selected from V_r, such that it maximizes the objective function:

$$G(v,\omega) = \sigma\big(\alpha \cdot \text{heading}(v,\omega) + \beta \cdot \text{dist}(v,\omega) + \gamma \cdot \text{velocity}(v,\omega)\big)$$

→ This is done by discritizing the Search Space.

★ Target heading

→ Measure of alignment of the robot with the target direction.

→ It is given by $(180 - \underbrace{\theta}_{\text{angle of the target point relative to robot heading direction}})$

⇒ θ is calculated for predicted position of the robot.

* Clearance

⇒ The function dist (V, ω) represent the distance to the closest obstacle that intersects with the curvature.

↳ If no obstacle is on the curvature the value is set to a large constant.

* Velocity

⇒ The function velocity (v, ω) is used to evaluate the progress of the robot on the corresponding trajectory.

* Smoothing

⇒ All three components of the objective function are normalized to [0,1].

↳ The weighted sum of the three components are calculated.
