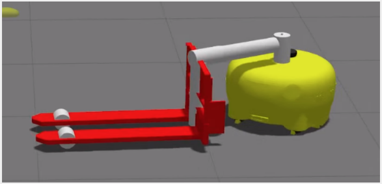
Autonomous Mobile Robotics

Motion Planning and Control

Geesara Kulathunga

September 7, 2022



Control of Mobile Robots

Contents

- Kinematics of wheeled mobile robots: internal, external, direct, and inverse
  - Differential drive kinematics
  - Bicycle drive kinematics
  - Rear-wheel bicycle drive kinematics
  - Car(Ackermann) drive kinematics
- Wheel kinematics constraints: rolling contact and lateral slippage
- Wheeled Mobile System Control: pose and orientation
  - Control to reference pose
  - Control to reference pose via an intermediate point
  - Control to reference pose via an intermediate direction
  - Control by a straight line and a circular arc
  - Reference path control
- Smooth path planning in a given 2-D space for vehicles with nonholonomic constraints using Hybrid A*

According to Fig. **??**,

- Steering angle α, steering wheel angular velocity ω_S, ICR point is defined by intersection of both wheel axes, and distance between wheels d

According to Fig. **??**,

- Steering angle α, steering wheel angular velocity ω_S, ICR point is defined by intersection of both wheel axes, and distance between wheels d
- We can define R(t)

$$R(t) = d\ tan(\frac{\pi}{2} - \alpha(t)) = \frac{d}{tan(\alpha(t))} \tag{1}$$

According to Fig. **??**,

- Steering angle α, steering wheel angular velocity ω_S, ICR point is defined by intersection of both wheel axes, and distance between wheels d
- We can define R(t)

$$R(t) = d\ tan(\frac{\pi}{2} - \alpha(t)) = \frac{d}{tan(\alpha(t))} \tag{1}$$

- Angular velocity ω around ICR

$$\omega(t) = \dot{\Phi} = \frac{\mathbf{v}_S(t)}{\sqrt{d^2 + R^2}} = \frac{v_S(t)}{d} sin(\alpha(t)) \tag{2}$$

According to Fig. **??**,

- Steering angle α, steering wheel angular velocity ω_S, ICR point is defined by intersection of both wheel axes, and distance between wheels d
- We can define R(t)

$$R(t) = d\ tan(\frac{\pi}{2} - \alpha(t)) = \frac{d}{tan(\alpha(t))} \tag{1}$$

- Angular velocity ω around ICR

$$\omega(t) = \dot{\Phi} = \frac{\mathbf{v}_S(t)}{\sqrt{d^2 + R^2}} = \frac{v_S(t)}{d} sin(\alpha(t)) \tag{2}$$

- Steering wheel velocity

$$\mathbf{v}_S(t) = \omega_S(t) r \tag{3}$$

- Internal robot kinematics

$$
\begin{aligned}
\dot{x}_m(t) &= \mathbf{v}_S(t) cos(\alpha(t)) \\
\dot{y}_m(t) &= 0 \\
\Phi\dot{(t)} &= \frac{\mathbf{v}_S(t)}{d} sin(\alpha(t)
\end{aligned}
\tag{4}
$$

- Internal robot kinematics

$$\begin{aligned} \dot{x}_m(t) &= \mathbf{v}_S(t)cos(\alpha(t)) \\ \dot{y}_m(t) &= 0 \\ \dot{\Phi(t)} &= \frac{\mathbf{v}_S(t)}{d}sin(\alpha(t) \end{aligned} \tag{4}$$

- External robot kinematics

$$\begin{aligned} \dot{x}(t) &= \mathbf{v}_S(t)cos(\alpha(t))cos(\Phi(t)) \\ \dot{y}(t) &= \mathbf{v}_S(t)cos(\alpha(t))sin(\Phi(t)) \\ \dot{\Phi(t)} &= \frac{\mathbf{v}_S(t)}{d}sin(\alpha(t) \end{aligned} \tag{5}$$

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\Phi(t)} \end{bmatrix} = \begin{bmatrix} cos(\Phi(t)) & 0 \\ sin(\Phi(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}(t) \\ \omega(t) \end{bmatrix} \tag{6}$$

, where $\mathbf{v}(t) = \mathbf{v}_S(t)cos(\alpha(t))$ and $\omega(t) = \frac{\mathbf{v}_S}{d}sin(\alpha(t))$

Motion control of Rear-Wheel Bicycle mobile robots

- Internal robot kinematics

$$
\begin{aligned}
\dot{x}_m(t) = \mathbf{v}_S(t)cos(\alpha(t)) = \mathbf{v}_r(t) \\
\dot{y}_m(t) = 0 \\
\dot{\Phi(t)} = \frac{\mathbf{v}_r(t)}{d}tan(\alpha(t)
\end{aligned}
\tag{7}
$$

Motion control of Rear-Wheel Bicycle mobile robots

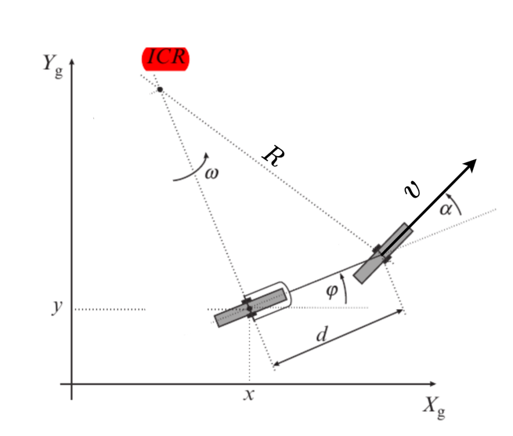
- Internal robot kinematics

$$\begin{aligned} \dot{x}_m(t) = \mathbf{v}_S(t)cos(\alpha(t)) = \mathbf{v}_r(t) \\ \dot{y}_m(t) = 0 \\ \dot{\Phi(t)} = \frac{\mathbf{v}_r(t)}{d}tan(\alpha(t) \end{aligned} \tag{7}$$

- External robot kinematics

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\Phi}(t) \end{bmatrix} = \begin{bmatrix} cos(\Phi(t)) & 0 \\ sin(\Phi(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_r(t) \\ \omega(t) \end{bmatrix} \tag{8}$$

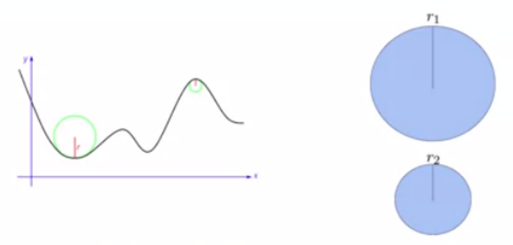
, where $\omega(t) = \frac{\mathbf{v}_r}{d}tan(\alpha(t))$


ICR
Y_g
R
ω
v
α
φ
y
d
x
X_g

- External robot kinematics

$$\begin{aligned} \dot{x}(t) &= v \cdot cos(\Phi(t) + \alpha(t)) \\ \dot{y}(t) &= v \cdot sin(\Phi(t) + \alpha(t)) \\ \dot{\Phi}(t) &= v/R = v/(d/sin(\alpha)) = v \cdot sin(\alpha)/d \\ \dot{\alpha} &= \text{input (rate of change of steering angle)} \end{aligned} \tag{9}$$

Motion control of Rear-Wheel Bicycle mobile robots

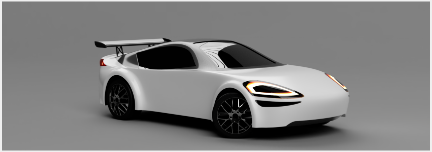


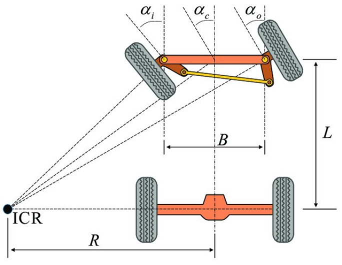
- Bicycle model imposes curvature constraint, where curvature is defined by

$$k = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{\left(\dot{x}^2(t) + \dot{y}^2(t)\right)^{3/2}}$$

- Curvature constraint is non-holonomic $v^2 \leq \frac{a_{lat}}{k}$, where $a_{lat} \leq a_{lat_{max}}$

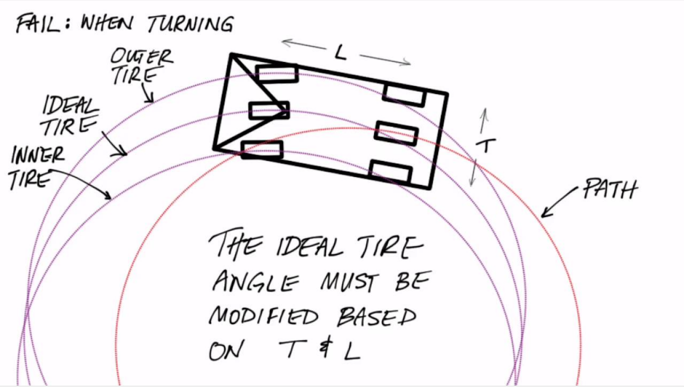
Motion control of Car(Ackermann) drive mobile robots





https://github.com/winstxnhdw/AutoCarROS2, https://doi.org/10.3390/s19214816

Motion control of Car(Ackermann) drive mobile robots



https://www.youtube.com/watch?v=i6uBwudwA5o

Motion control of Car(Ackermann) drive mobile robots

- Uses steering principle, i.e., inner wheel, which is closer to its ICR, should steer for a bigger angle than the outer wheel, Consequently the inner wheel travels with slower speed than the outer wheel

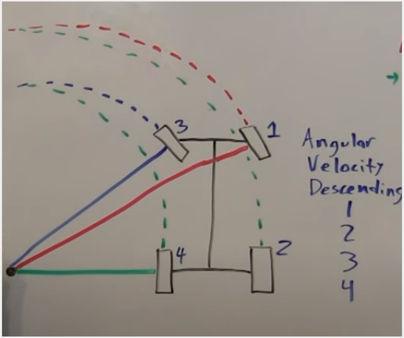


Figure: Angular velocity speed descending order

- Ackermann geometry is to avoid the need for tires to slip sideways when following the path around a curve which requires that the ICR point lies on a straight line defined by the rear wheels' axis

- Ackermann geometry is to avoid the need for tires to slip sideways when following the path around a curve which requires that the ICR point lies on a straight line defined by the rear wheels' axis
- Ackermann geometry can be seen as two bicycles welded together side by side

Motion control of Car(Ackermann) drive mobile robots

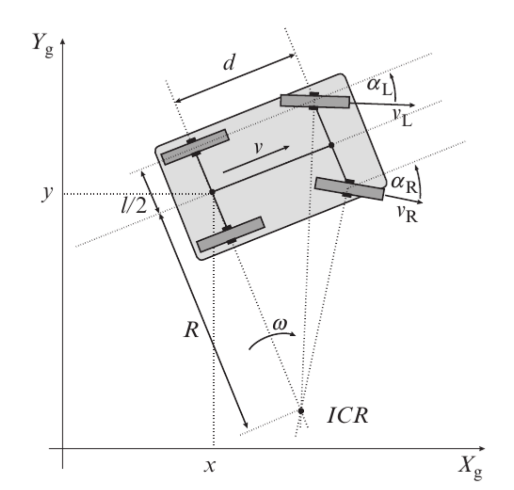
- Ackermann geometry is to avoid the need for tires to slip sideways when following the path around a curve which requires that the ICR point lies on a straight line defined by the rear wheels' axis
- Ackermann geometry can be seen as two bicycles welded together side by side
- For differential drive it needs individual drives at each wheel which makes the system more complex

Motion control of Car(Ackermann) drive mobile robots

- Ackermann geometry is to avoid the need for tires to slip sideways when following the path around a curve which requires that the ICR point lies on a straight line defined by the rear wheels' axis
- Ackermann geometry can be seen as two bicycles welded together side by side
- For differential drive it needs individual drives at each wheel which makes the system more complex
- Ackerman steering adjusts the relative angles of the steerable wheels so they both run true around a curve. Differentials allow the two driven wheels to run at different speeds around a curve, quite a different requirement

Motion control of Car(Ackermann) drive mobile robots

Motion control of Car(Ackermann) drive mobile robots



- Steering wheels orientations

$$
\begin{aligned}
tan(\frac{\pi}{2} - \alpha_L) = \frac{R + l/2}{d} \rightarrow \alpha_L = \frac{\pi}{2} - arctan\Big(\frac{R + l/2}{d}\Big) \\
tan(\frac{\pi}{2} - \alpha_R) = \frac{R - l/2}{d} \rightarrow \alpha_R = \frac{\pi}{2} - arctan\Big(\frac{R - l/2}{d}\Big)
\end{aligned}
\tag{10}
$$

Motion control of Car(Ackermann) drive mobile robots

- Steering wheels orientations

$$
\begin{aligned}
tan(\frac{\pi}{2} - \alpha_L) = \frac{R + l/2}{d} \rightarrow \alpha_L = \frac{\pi}{2} - arctan\Big(\frac{R + l/2}{d}\Big) \\
tan(\frac{\pi}{2} - \alpha_R) = \frac{R - l/2}{d} \rightarrow \alpha_R = \frac{\pi}{2} - arctan\Big(\frac{R - l/2}{d}\Big)
\end{aligned}
\tag{10}
$$

- Back wheels (inner and outer) velocities

$$
\begin{aligned}
\mathbf{v}_L = \omega(R + \frac{l}{2}) \\
\mathbf{v}_R = \omega(R - \frac{l}{2})
\end{aligned}
\tag{11}
$$

Motion control of Car(Ackermann) drive mobile robots

- Steering wheels orientations

$$
\begin{aligned}
tan(\frac{\pi}{2} - \alpha_L) = \frac{R + l/2}{d} \rightarrow \alpha_L = \frac{\pi}{2} - arctan\left(\frac{R + l/2}{d}\right) \\
tan(\frac{\pi}{2} - \alpha_R) = \frac{R - l/2}{d} \rightarrow \alpha_R = \frac{\pi}{2} - arctan\left(\frac{R - l/2}{d}\right)
\end{aligned} \tag{10}
$$

- Back wheels (inner and outer) velocities

$$
\begin{aligned}
\mathbf{v}_L = \omega(R + \frac{l}{2}) \\
\mathbf{v}_R = \omega(R - \frac{l}{2})
\end{aligned} \tag{11}
$$

- Inverse kinematics is quite complicated (TODO)

Define Mobile Robots with Kinematic Constraints

- **Unicycle Kinematic Model** The simplest way to represent mobile robot vehicle kinematics is with a unicycle model, which has a wheel speed set by a rotation about a central axle, and can pivot about its z-axis. Both the differential-drive and bicycle kinematic models reduce down to unicycle kinematics when inputs are provided as vehicle speed and vehicle heading rate and other constraints are not considered.

https://nl.mathworks.com/help/robotics/ref/ackermannkinematics.html

Define Mobile Robots with Kinematic Constraints

- **Unicycle Kinematic Model** The simplest way to represent mobile robot vehicle kinematics is with a unicycle model, which has a wheel speed set by a rotation about a central axle, and can pivot about its z-axis. Both the differential-drive and bicycle kinematic models reduce down to unicycle kinematics when inputs are provided as vehicle speed and vehicle heading rate and other constraints are not considered.
- **Differential-Drive Kinematic Model** uses a rear driving axle to control both vehicle speed and head rate. The wheels on the driving axle can spin in both directions. Since most mobile robots have some interface to the low-level wheel commands, this model will again use vehicle speed and heading rate as input to simplify the vehicle control.

https://nl.mathworks.com/help/robotics/ref/ackermannkinematics.html

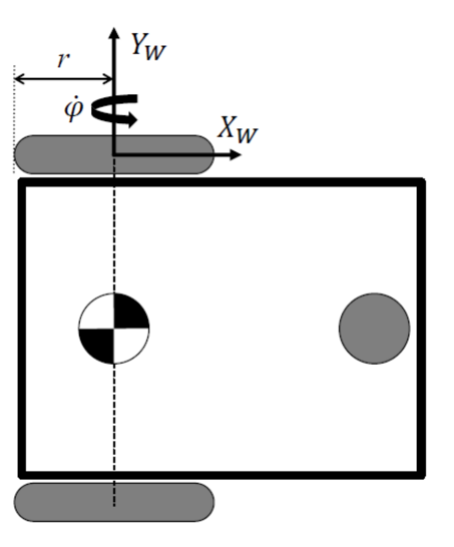
Define Mobile Robots with Kinematic Constraints

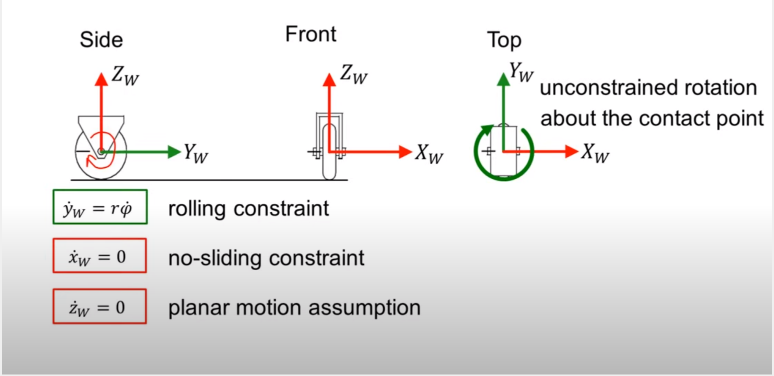
- **Bicycle Kinematic Model** treats the robot as a car-like model with two axles: a rear driving axle, and a front axle that turns about the z-axis. The bicycle model works under the assumption that wheels on each axle can be modeled as a single, centered wheel, and that the front wheel heading can be directly set, like a bicycle.

Define Mobile Robots with Kinematic Constraints

- **Bicycle Kinematic Model** treats the robot as a car-like model with two axles: a rear driving axle, and a front axle that turns about the z-axis. The bicycle model works under the assumption that wheels on each axle can be modeled as a single, centered wheel, and that the front wheel heading can be directly set, like a bicycle.
- **Ackermann kinematic model** is a modified car-like model that assumes Ackermann steering. In most car-like vehicles, the front wheels do not turn about the same axis, but instead turn on slightly different axes to ensure that they ride on concentric circles about the center of the vehicle's turn. This difference in turning angle is called Ackermann steering, and is typically enforced by a mechanism in actual vehicles. From a vehicle and wheel kinematics standpoint, it can be enforced by treating the steering angle as a rate

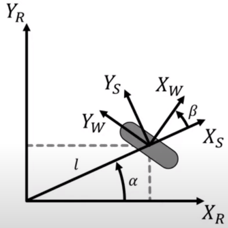
Wheel kinematics constraints





https://www.youtube.com/watch?v=hu__jYsN6mw

Wheel kinematics constraints



robot angular velocity

$$\mathbf{v}_{IW} = \mathbf{v}_{IR} + \boldsymbol{\omega}_{IR} \times \mathbf{r}_{RS}$$

robot velocity

steering offset

Coordinate frames

- I – inertial
- R – robot
- S – steering
- W – wheel

https://www.youtube.com/watch?v=hu__jYsN6mw

Wheel kinematics constraints

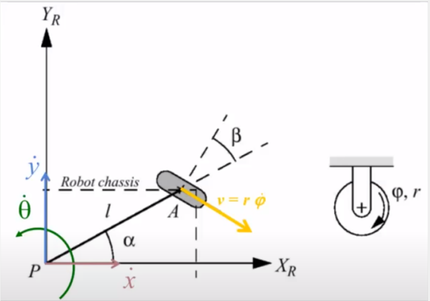
There are different types of wheel types, each of which has own constraints. For this course we only focus on standard wheel type. The following important assumptions are made

- Plane of wheel always remains vertical, where only one single point of contact between the and ground plane
- No sliding at this single point of contact

$$\begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \begin{bmatrix}\dot{\varphi} r\\ 0\end{bmatrix}\begin{array}{l}\text{Rolling constraint}\\ \text{No-sliding constraint}\end{array}$$

Wheel kinematics constraints: Fixed standard wheel

- α, β, and l locate the relative to the robot internal (local) frame
- θ is the angle between inertial x-axis and X_R (global frame)
- What differential constraints on velocity does the wheel impose on the chassis?



https://asl.ethz.ch/education/lectures/autonomous_mobile_rob

Wheel kinematics constraints: Fixed standard wheel

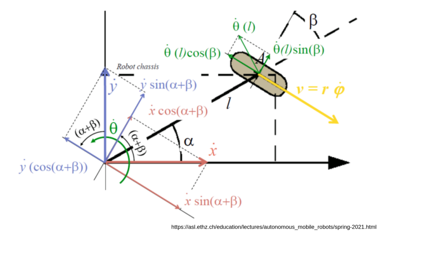
Two constraints can be derived based on those assumptions. The position A is expressed in polar coordinates by distance l and angle α

- rolling contact

$$[sin(\alpha+\beta) \quad -cos(\alpha+\beta) \quad (-l)cos(\beta)][\dot{x}\ \dot{y}\ \dot{\theta}]^\top - r\dot{\Phi} = 0 \tag{12}$$

- no lateral slippage

$$[cos(\alpha+\beta) \quad sin(\alpha+\beta) \quad (l)sin(\beta)][\dot{x}\ \dot{y}\ \dot{\theta}]^\top = 0 \tag{13}$$



- Can control start pose to goal pose by classical control, where intermediate state trajectory is not prescribed or reference trajectory tracking

- Can control start pose to goal pose by classical control, where intermediate state trajectory is not prescribed or reference trajectory tracking
- Nonholonomic constraints need to be considered, in such occasion, the controller is is twofold: feedforward control and feedback control, namely two-dgree-of-freedom control.

- Can control start pose to goal pose by classical control, where intermediate state trajectory is not prescribed or reference trajectory tracking
- Nonholonomic constraints need to be considered, in such occasion, the controller is is twofold: feedforward control and feedback control, namely two-dgree-of-freedom control.
- Openloop control: feedforward control is calculated from the reference trajectory and those control action are fed to system

- Can control start pose to goal pose by classical control, where intermediate state trajectory is not prescribed or reference trajectory tracking
- Nonholonomic constraints need to be considered, in such occasion, the controller is is twofold: feedforward control and feedback control, namely two-dgree-of-freedom control.
- Openloop control: feedforward control is calculated from the reference trajectory and those control action are fed to system
- However, feedforward control is not practical as it is not robust to disturbance, feedback needs to be applied

Wheeled Mobile System Control

- Can control start pose to goal pose by classical control, where intermediate state trajectory is not prescribed or reference trajectory tracking
- Nonholonomic constraints need to be considered, in such occasion, the controller is is twofold: feedforward control and feedback control, namely two-dgree-of-freedom control.
- Openloop control: feedforward control is calculated from the reference trajectory and those control action are fed to system
- However, feedforward control is not practical as it is not robust to disturbance, feedback needs to be applied
- Wheeled mobile robots are dynamic. Thus, motion controlling system has to incorporate dynamics of the system, in general, which systems are designed as cascade control schemes: outer controller for velocity control and inner controller to handle torque, force, etc.

Target (reference) pose control

- Pose = position + orientation

- Pose = position + orientation
- Feasible path, which can be **optimal**, should satisfy the kinematic, dynamic, and other constraints including disturbances, appropriately

Target (reference) pose control

- Pose = position + orientation
- Feasible path, which can be **optimal**, should satisfy the kinematic, dynamic, and other constraints including disturbances, appropriately
- Reference pose control, in general, is performed as two sub controlling tasks: orientation control and forward-motion control. However, these are interconnected each other

Target (reference) Orientation Control

- Orientation control cannot be performed independently from the forward-motion control

- Orientation control cannot be performed independently from the forward-motion control
- Let robot orientation $\Phi(t)$, at time t, and reference orientation is $\Phi_{ref}(t)$

$$e_{\Phi}(t) = \Phi_{ref}(t) - \Phi(t) \tag{14}$$

Target (reference) Orientation Control

- Orientation control cannot be performed independently from the forward-motion control
- Let robot orientation $\Phi(t)$, at time t, and reference orientation is $\Phi_{ref}(t)$

$$e_{\Phi}(t) = \Phi_{ref}(t) - \Phi(t) \tag{14}$$

- How fast can we drive the control error to zero? It depends on additional factors: energy consumption, actuator load, and robustness

Target (reference) Orientation Control

- Orientation control cannot be performed independently from the forward-motion control
- Let robot orientation $\Phi(t)$, at time t, and reference orientation is $\Phi_{ref}(t)$

$$e_{\Phi}(t) = \Phi_{ref}(t) - \Phi(t) \tag{14}$$

- How fast can we drive the control error to zero? It depends on additional factors: energy consumption, actuator load, and robustness
- Since $\dot{\Phi}(t) = \omega(t)$ is the input for control for diff drive, a proportional controller is able to drive control error of an integral process to 0

$$\omega(t) = K(\Phi_{ref} - \Phi(t)) \tag{15}$$

, where K is an arbitrary positive constant

Target (reference) Orientation Control

- $\dot{\Phi}(t) = \frac{\mathbf{v}_r}{d} tan(\alpha(t))$ is the input for control for Ackermann drive. The control variable is α, which can be chosen proportional to the orientation error:

$$\begin{aligned} \alpha(t) &= K\,(\Phi_{ref}(t) - \Phi(t)) \\ \dot{\Phi}(t) &= \frac{\mathbf{v}_r}{d} tan(K\,(\Phi_{ref}(t) - \Phi(t))) \end{aligned} \tag{16}$$

- $\dot{\Phi}(t) = \frac{\mathbf{v}_r}{d} tan(\alpha(t))$ is the input for control for Ackermann drive. The control variable is α, which can be chosen proportional to the orientation error:

$$\begin{aligned} \alpha(t) &= K\left(\Phi_{ref}(t) - \Phi(t)\right) \\ \dot{\Phi}(t) &= \frac{\mathbf{v}_r}{d} tan(K\left(\Phi_{ref}(t) - \Phi(t)\right)) \end{aligned} \tag{16}$$

- For small angle and constant velocity of rear wheels $\mathbf{v}_r(t) = V$, a linear approximation can be obtained,

$$\dot{\Phi}(t) = \frac{V}{d}(K\left(\Phi_{ref}(t) - \Phi(t)\right)) \tag{17}$$

- Forward-motion control is inevitably interconnected with orientation control, i.e., forward-motion alone can not drive to goal pose without correct orientation

$$\mathbf{v}(\mathbf{t}) = K\sqrt{((x_{ref}(t) - x(t))^2 + (y_{ref}(t) - y(t))^2)} \tag{18}$$

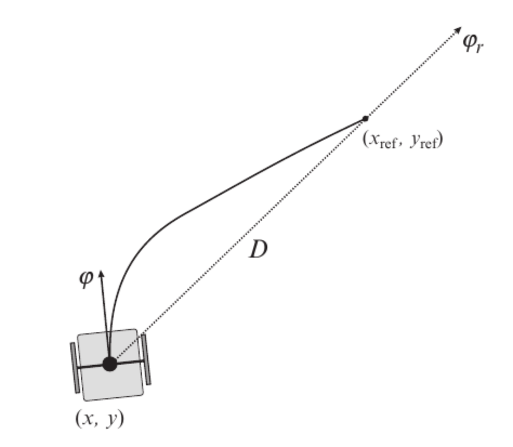
Target (reference) Forward-motion Control

- Forward-motion control is inevitably interconnected with orientation control, i.e., forward-motion alone can not drive to goal pose without correct orientation

$$\mathbf{v}(\mathbf{t}) = K\sqrt{((x_{ref}(t) - x(t))^2 + (y_{ref}(t) - y(t))^2)} \tag{18}$$

- However, $\mathbf{v}(t)$ should have maximum limits, which is due to actuator limitations driving surface conditions. On the other hand, when robot get closer to goal, it might try to over take the reference pose, which is eventually lead to accelerate, which is not desired

Control to reference pose



- It is required to reach to the target position where the final orientation is not prescribed, hence direction of reference position

$$
\begin{aligned}
\Phi_r(t) = arctan\frac{y_{ref} - y(t)}{x_{ref} - x(t)}, \omega(t) = K_1(\Phi_r(t) - \Phi(t)) \\
\mathbf{v}(\mathbf{t}) = K\sqrt{((x_{ref}(t) - x(t))^2 + (y_{ref}(t) - y(t))^2)}
\end{aligned}
\tag{19}
$$

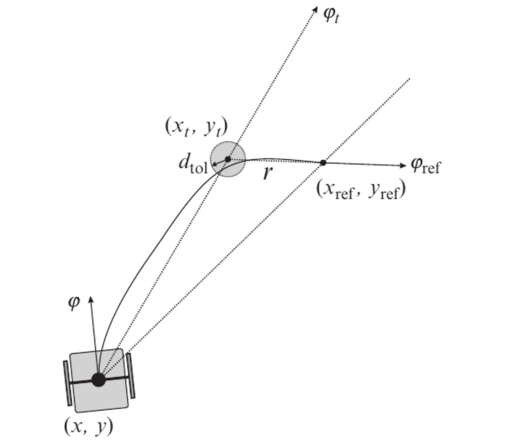
- It is required to reach to the target position where the final orientation is not prescribed, hence direction of reference position

$$\Phi_r(t) = arctan\frac{y_{ref} - y(t)}{x_{ref} - x(t)}, \omega(t) = K_1(\Phi_r(t) - \Phi(t))$$
$$\mathbf{v}(\mathbf{t}) = K\sqrt{((x_{ref}(t) - x(t))^2 + (y_{ref}(t) - y(t))^2)} \tag{19}$$

- What will happen when orientation error abruptly changes ($\pm$ 180 degrees)? if the absolute value of orientation error exceeds 90 degree, orientation error increased or decreased by 180 degree

$$e_\Phi(t) = \Phi_{ref}(t) - \Phi(t), \omega(t) = K_1 arctan(tan(e_\Phi(t)))$$
$$\mathbf{v}(\mathbf{t}) = K\sqrt{((x_{ref}(t) - x(t))^2 + (y_{ref}(t) - y(t))^2)}.sgn(cos(e_\Phi(t))) \tag{20}$$

Control to reference pose via an intermediate point



Control to reference pose via an intermediate point

- Idea is to shape in a way that the correct orientation is obtained

Control to reference pose via an intermediate point

- Idea is to shape in a way that the correct orientation is obtained
- Intermediate point is determined by

$$\begin{aligned} x_t &= x_{ref} - r\, cos(\Phi_{ref}) \\ y_t &= y_{ref} - r\, sin(\Phi_{ref}) \end{aligned} \tag{21}$$

, where distance from reference point to intermediate point denoted r

Control to reference pose via an intermediate point

- Idea is to shape in a way that the correct orientation is obtained
- Intermediate point is determined by

$$
\begin{aligned}
x_t &= x_{ref} - r\, cos(\Phi_{ref}) \\
y_t &= y_{ref} - r\, sin(\Phi_{ref})
\end{aligned}
\tag{21}
$$

  , where distance from reference point to intermediate point denoted r
- If distance between current and intermediate position $\sqrt{(x - x_t)^2 + (y - y_t)^2} < d_{tol}$, where term d_{tol} depicts threshold, robot starts controlling to reference point

References

Gregor Klancar, Andrej Zdesar, Saso Blazic, and Igor Skrjanc.
Wheeled mobile robotics: from fundamentals towards autonomous systems.
Butterworth-Heinemann, 2017.

Roland Siegwart, Illah Reza Nourbakhsh, and Davide Scaramuzza.

Introduction to autonomous mobile robots.
MIT press, 2011.

Sebastian Thrun.
Probabilistic robotics.
Communications of the ACM, 45(3):52–57, 2002.