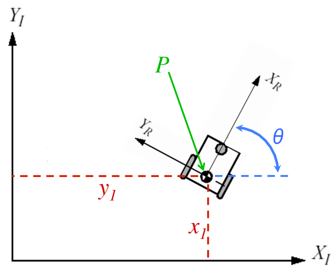
Kinematics
Transforms & Wheels



Class Today

2

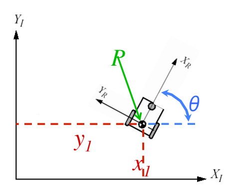
- 2D transformations
- Rolling and

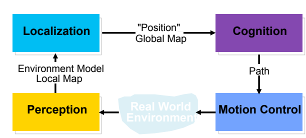
Project Next Steps

4

- By now you should have:
  - Built robot
  - Installed Raspbian
- Next important step: what will your architecture be?
  - Code and version control?
  - Message passing and comms infrastructure?
- Turnins
  - Writeup of architecture
  - Code to control servos and read sensors
  - Video of a small demo

Kinematics
Mobile Kinematics

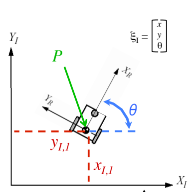




Mobile Kinematics: Concepts

8

- Forward Kinematics:
  - Parameters → Configuration
- Inverse Kinematics (IK):
  - Configuration → Parameters
  - I want to be in this configuration. What motions should I make?

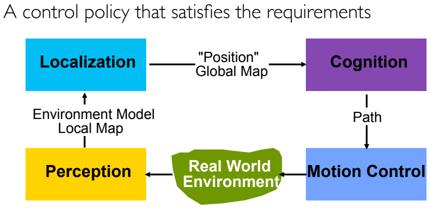


- Mobile configuration = position and orientation with respect to an arbitrary initial frame I
- Understanding mobile robot motion starts with understanding constraints on the robot's mobility.

Wheeled Motion Control

9

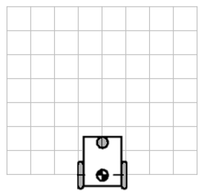
- Requirements for understanding/controlling motion:
  1. Kinematic / dynamic model of the robot
  2. Model of interaction between the wheel and the ground
  3. Definition of required motion
     - What speed and position controls are there? Are possible?
  4. A control policy that satisfies the requirements



What We're Trying to Do

10

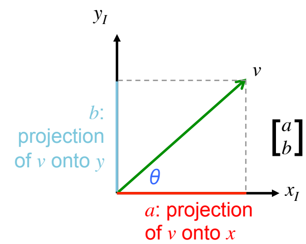
- Sequence of events:
  1. Power on: position = (0,0), orientation = due north
  2. Rotate 15° right
  3. Move forward 2 meters
  4. Observe obstacle
  5. Rotate 30° left
  6. Move forward 1 meter
- Position – (?,?), orientation = ?°
- Where's the obstacle?



Projections

12

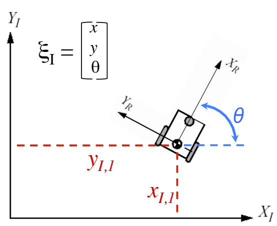
- A projection of a vector v onto an axis is the amount of change along that axis along the length of v.
- This is the change in position in that axis
- Here:
  - $a = \Delta x_I$
  - $b = \Delta y_I$
- This is the change in position in that axis



Specifying Transforms

13

- How does a robot (or system) map to the global frame of reference?
- Configuration = position and orientation
- Position: x,y coordinates $x_{I,t}$ and $y_{I,t}$
  - I = initial (global)
  - t = timestep
- Orientation: θ
  - Angle between robot's coordinate system and initial coordinate system

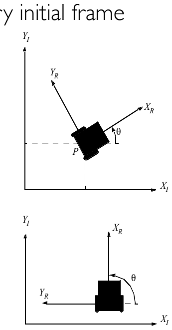


Mapping Between Frames

17

- Representing robot within an arbitrary initial frame
  - Initial frame: $\{X_I, Y_I\}$
  - Robot frame: $\{X_R, Y_R\}$
  - Robot: $\xi_I = [x \quad y \quad \theta]^T$
    - Just the transpose
- Goal
  - Map motions from global reference frame to local reference frame (and vice versa)

$$\{X_I, Y_I\} \longrightarrow \{X_R, Y_R\}$$



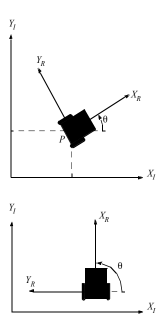
Mapping Between Frames

18

- Global reference frame ←→ local reference frame

$$\{X_I, Y_I\} \longleftrightarrow \{X_R, Y_R\}$$

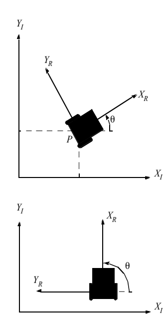
  - Map motion from **axes** of one to **axes** of the other
    - This mapping depends on current pose
- How do you do this mapping?
- How do you perform a rotation in Euclidean spaces?



Mapping Between Frames

19

- How do you perform a rotation?
- A **rotation matrix** is used to perform a rotation in Euclidean space.
- Any point <x,y> in space (aka $\begin{bmatrix} x \\ y \end{bmatrix}$) can be multiplied by some matrix...
  - (spoiler: it's → $R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$ usually)
- The result is the coordinates in the other frame, rotated by θ around z.
  - This matrix rotates points in the xy plane counter-clockwise, through θ, around the origin.



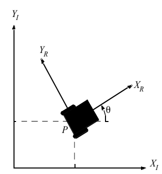
Mapping Between Frames

20

- Global reference frame ⟷ local reference frame

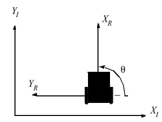
  $\{X_I, Y_I\} \leftrightarrow \{X_R, Y_R\}$

  - Map motion from **axes** of one to **axes** of the other
    - This mapping depends on current pose



- Use *orthogonal reference frame**:

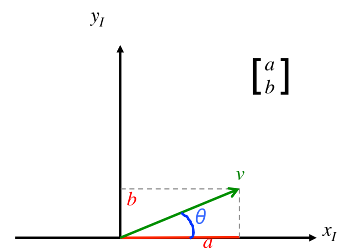
$$R(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$



**this one rotates around z*

The Z Rotation Matrix

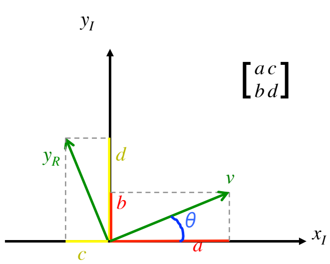
21



* AKA *orthogonal rotation matrix*

The Z Rotation Matrix*

22

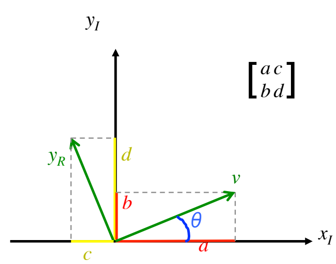


* AKA *orthogonal rotation matrix*

The Z Rotation Matrix

23

- If we assume frame axes are of length 1…



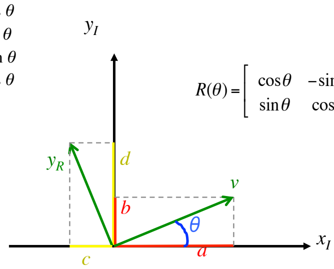
* AKA *orthogonal rotation matrix*

The Z Rotation Matrix

24

- If we assume frame axes are of length 1
  - $a = \cos\theta$
  - $b = \sin\theta$
  - $c = -\sin\theta$
  - $d = \cos\theta$

$$R(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$$



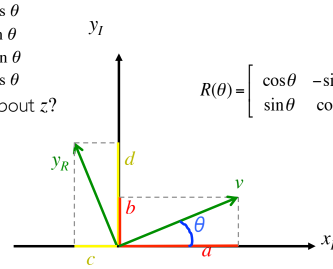
* AKA *orthogonal rotation matrix*

The Z Rotation Matrix

25

- If we assume frame axes are of length 1
  - $a = \cos\theta$
  - $b = \sin\theta$
  - $c = -\sin\theta$
  - $d = \cos\theta$
- What about *z*?

$$R(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}$$



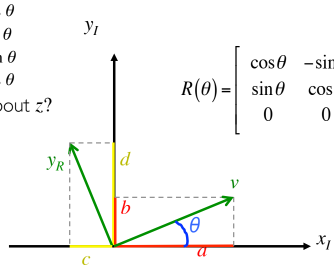
* AKA *orthogonal rotation matrix*

The Z Rotation Matrix

26

- If we assume frame axes are of length 1
  - $a = \cos\theta$
  - $b = \sin\theta$
  - $c = -\sin\theta$
  - $d = \cos\theta$
- What about z?

$$R(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$



* AKA *orthogonal rotation matrix*

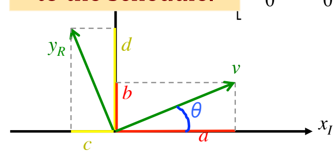
The Z Rotation Matrix

27

- If we assume frame axes are of length 1
  - $a = \cos\theta$
  - $b = \sin\theta$
  - $c = -\sin\theta$
  - $d = \cos\theta$
- What about z?

Some really useful videos are posted to the schedule.

$$\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$



* AKA *orthogonal rotation matrix*

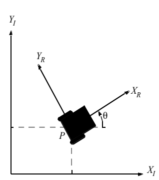
Mapping Between Frames

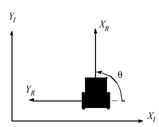
28

- How do you perform a rotation, again?
- A **rotation matrix** is used to perform a rotation in Euclidean space.

$$\begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\begin{bmatrix}x_R\\ y_R\end{bmatrix}=\begin{bmatrix}x_I\\ y_I\end{bmatrix}$$

- Rotates points in the xy plane counter-clockwise, through θ, around the origin.
- To use R, the position of **each point** is represented by a vector.
- A rotated vector is then obtained with matrix multiplication.





Orthogonal Rotation Matrix

29

- This mapping function is called

  $R(\theta)\,\dot{\xi}_I$

  because it depends on θ.

$$R(\theta)=\begin{bmatrix}\cos\theta & -\sin\theta & 0\\ \sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

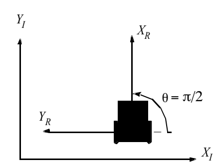
Z rotation matrix

$\dot{\xi}_R = R(\pi/2)\,\dot{\xi}_I$

- Example:

$$R(\pi/2)=\begin{bmatrix}0 & 1 & 0\\ -1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}$$

Why?



cos(pi/2)=0
sin(pi/2)=1
tan(pi/2)=infty
cot(pi/2)=0
csc(pi/2)=1
sec(pi/2)=infty

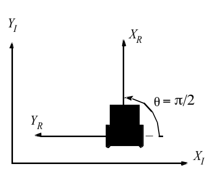
Velocity Vector

30

- Given some velocity in I:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix}$$

- We can compute motion along X_R and Y_R.
  - (Or vice versa.)
- This example:
  - Motion along $X_R = \dot{y}$
  - Motion along $Y_R = -\dot{x}$



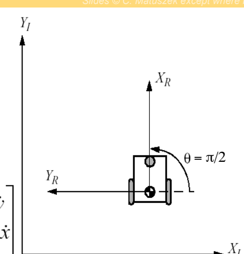
Example, cont'd

31

No longer around Z!

$$R(\theta)=\begin{bmatrix}\cos\theta & \sin\theta & 0\\ -\sin\theta & \cos\theta & 0\\ 0 & 0 & 1\end{bmatrix}$$

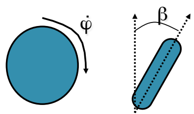
$$\dot{\xi}_R = R(\frac{\pi}{2})\dot{\xi}_I = \begin{bmatrix}0 & 1 & 0\\ -1 & 0 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}\dot{y}\\ -\dot{x}\\ \dot{\theta}\end{bmatrix}$$



Kinematics Models

33

- Goal:
  - Establish speed $\dot{\xi} = [\dot{x} \;\; \dot{y} \;\; \dot{\theta}]^T$ as a function of the wheel speeds $\dot{\varphi}_i$, steering angles β_i, steering speeds $\dot{\beta}_i$ and the geometric parameters of the robot (configuration coordinates)



- $\dot{\varphi}$ measured in radians/sec, so $\dot{\varphi}/2\pi$ is revolutions/sec
- In one revolution wheel translates $2\pi r$ linear units
- Translational velocity is $2\pi r(\dot{\varphi}/2\pi) = r\dot{\varphi}$

Forward Kinematics Models

34

- Goal:
  - Establish speed $\dot{\xi} = [\dot{x} \;\; \dot{y} \;\; \dot{\theta}]^T$ as a function of the wheel speeds $\dot{\varphi}_i$, steering angles β_i, steering speeds $\dot{\beta}_i$ and the geometric parameters of the robot (configuration coordinates)
- Forward kinematics:

"If I do this, what will happen?"

$$\dot{\xi} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\dot{\varphi}_1, \cdots \dot{\varphi}_n, \beta_1, \cdots \beta_m, \dot{\beta}_1, \cdots \dot{\beta}_m)$$

Inverse Kinematics Models

35

- Goal:
  - Establish speed $\dot{\xi} = [\dot{x} \;\; \dot{y} \;\; \dot{\theta}]^T$ as a function of the wheel speeds $\dot{\varphi}_i$, steering angles β_i, steering speeds $\dot{\beta}_i$ and the geometric parameters of the robot (configuration coordinates)
- Inverse kinematics:

"If I want this to happen, what should I do?"

$$[\dot{\varphi}_1 \; \cdots \; \dot{\varphi}_n \; \beta_1 \; \cdots \; \beta_m \; \dot{\beta}_1 \; \cdots \; \dot{\beta}_m]^T = f(\dot{x}, \dot{y}, \dot{\theta})$$

Differential Drive Model

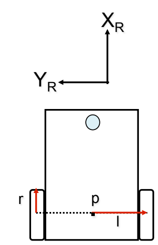
36

- The robot has:
  - Two wheels - radius *r*
  - Point P centered between wheels
  - Each wheel is distance l (ℓ) from P
  - Wheels have rotational velocity $\dot{\varphi}_1$ and $\dot{\varphi}_2$
- Forward kinematic model

$$\dot{\xi}_I = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = f(\ell, r, \theta, \dot{\varphi}_1, \dot{\varphi}_2)$$

- Mapping from global to local is

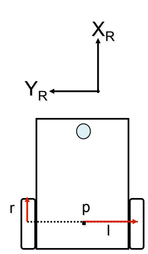
$$\dot{\xi}_R = R(\theta)\,\dot{\xi}_I \text{ , so } \dot{\xi}_I = R^{-1}(\theta)\,\dot{\xi}_R$$



Differential Drive (cont.)

37

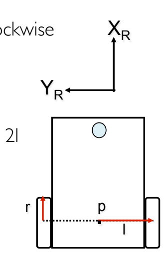
- Since $\dot{\xi}_R = R(\theta)\,\dot{\xi}_I$, $\dot{\xi}_I = R^{-1}(\theta)\,\dot{\xi}_R$
- Compute how wheel speeds influence $\dot{\xi}_R$
- Translate to $\dot{\xi}_I$ via $R^{-1}(\theta)$
- Contribution to translation along X_R
- If one wheel spins and the other is still:
  - P will move at half the translational velocity of the wheel: $1/2r\dot{\varphi}_1$ or $1/2r\dot{\varphi}_2$
  - Sum these for both wheels spinning
    - $\dot{X}_R = 1/2r\dot{\varphi}_1 + 1/2r\dot{\varphi}_2$
- What if they spin in opposite directions? Same direction?



Differential Drive (cont.)

38

- Wheel rotation never contributes to Y_R. Why?
- What about θ?
  - Wheel 1 spin makes robot rotate counterclockwise
  - Pivot around wheel 2 (left wheel)
  - Translational velocity is $r\dot{\varphi}$
  - Traces circle with radius 2l
  - Rotational velocity $2\pi * r\dot{\varphi} / (2\pi * 2l) = r\dot{\varphi} / 2l$
  - Wheel 2 spin makes robot rotate clockwise
  - Sum to get net effect: $\dot{\theta} = (r\dot{\varphi}_1 - r\dot{\varphi}_2) / 2l$



Differential Drive: The Punchline

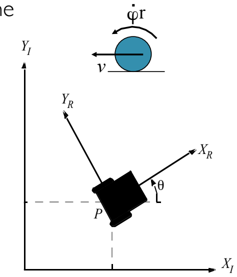
39

$$\dot{\xi}_I = R^{-1}(\theta)\,\dot{\xi}_R = R^{-1}(\theta)\begin{bmatrix} r(\dot{\varphi}_1 + \dot{\varphi}_2)/2 \\ 0 \\ r(\dot{\varphi}_1 - \dot{\varphi}_2)/2l \end{bmatrix}$$

Wheel Constraints: Assumptions

40

- Movement is on a horizontal plane
- Wheels:
  - Make point contact
  - Are not deformable
  - Are connected to rigid chassis
  - Have steering axes orthogonal to surface being moved on
- Constraints
  - Pure rolling
  - No slipping, skidding or sliding
  - No friction in rotation around contact point

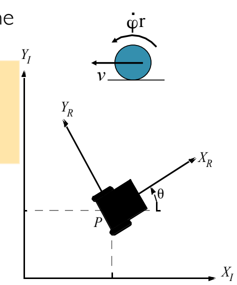


Wheel Constraints: Assumptions

41

- Movement is on a horizontal plane
- Wheels:
  - Make poi
  - Are not c
  - Are conr
  - Have steering axes orthogonal to surface being moved on
- Constraints
  - Pure rolling
  - No slipping, skidding or sliding
  - No friction in rotation around contact point

How do we represent these constraints?

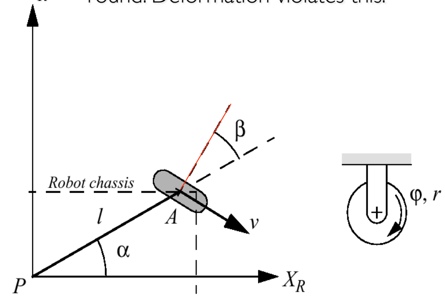


Wheels: Round Constraint

42

Round constraint: the wheel must be (perfectly) round. Deformation violates this.

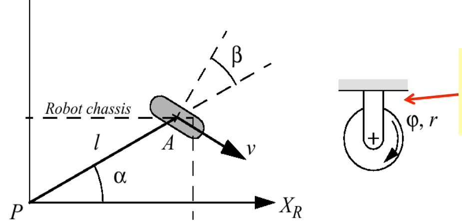
$$\dot{\xi}_I = \begin{bmatrix} \dot{x}_R \\ \dot{y}_R \\ \dot{\theta}_R \end{bmatrix}$$



Wheels: Rolling Constraint

43

Rolling constraint: all motion along wheel plane (**in the direction of v**) must be accompanied by the same amount of wheel spin so that there is pure rolling at contact point



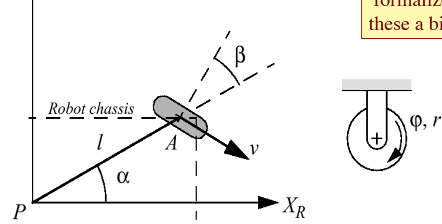
We're discussing *fixed wheel A*

Wheels: Sliding Constraint

44

Sliding constraint: there can be no motion orthogonal to wheel plane (perpendicular to v), otherwise wheel skids

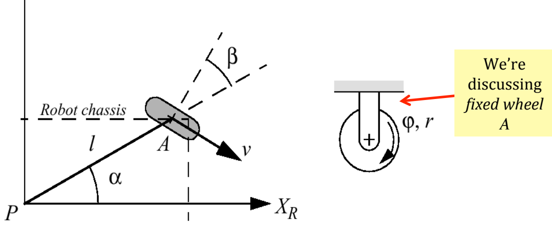
So let's formalize these a bit.



Wheels: Rolling Constraint (2)

Rolling constraint:

$$\left[\sin(\alpha+\beta)\;\; -\cos(\alpha+\beta)\;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

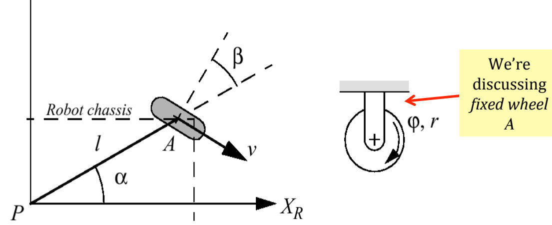


Wheels: Rolling Constraint (3)

Rolling constraint:

$$\left[\sin(\alpha+\beta)\;\; -\cos(\alpha+\beta)\;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

Total motion in wheel plane

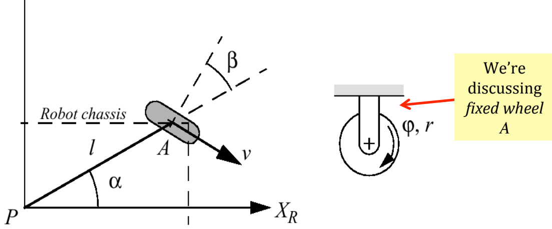


Wheels: Rolling Constraint

Rolling constraint:

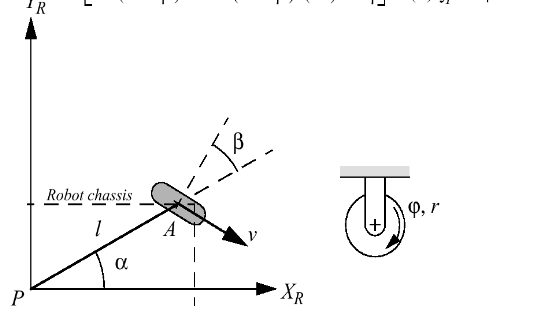
$$\left[\sin(\alpha+\beta)\;\; -\cos(\alpha+\beta)\;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

Same transformation: I → R



Wheels: Round Constraint

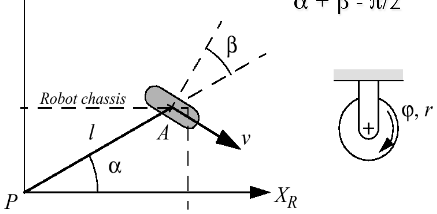
$$\left[\sin(\alpha+\beta)\;\; -\cos(\alpha+\beta)\;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$



Round Constraint (2)

$$\left[\sin(\alpha+\beta)\;\; -\cos(\alpha+\beta)\;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

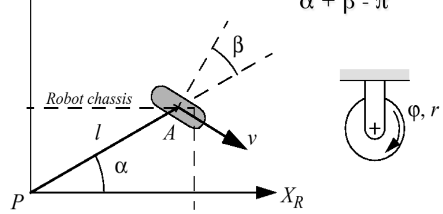
Angle between X_R and v is $\alpha + \beta - \pi/2$



Round Constraint (3)

$$\left[\sin(\alpha+\beta)\;\; -\cos(\alpha+\beta)\;\; (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

Angle between Y_R and v is $\alpha + \beta - \pi$



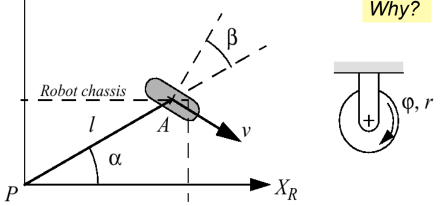
Round Constraint (4)

51

$$\left[\sin(\alpha+\beta)\ \ -\cos(\alpha+\beta)\ \ (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

When robot rotates, A has translational velocity $l\dot{\theta}$. *Why?*

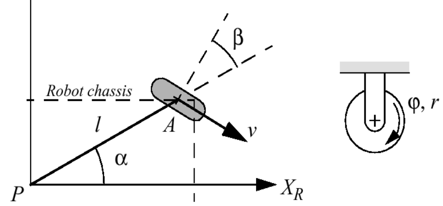
Component in direction of V is $-l\dot{\theta}\cos\beta$. *Why?*



Sliding Constraint

52

$$\left[\cos(\alpha+\beta)\ \ \sin(\alpha+\beta)\ \ l\sin\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$



Example

53

$$\left[\sin(\alpha+\beta)\ \ -\cos(\alpha+\beta)\ \ (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \quad \text{round}$$
$$\left[\cos(\alpha+\beta)\ \ \sin(\alpha+\beta)\ \ l\sin\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0 \quad \text{sliding}$$

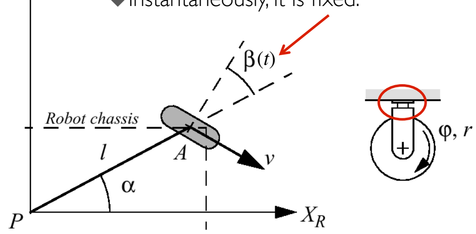
- Suppose that the wheel A is in position such that $\alpha = 0$ and $\beta = 0$
- Puts contact point of wheel on X_I, with plane of the wheel oriented parallel to Y_I
- If $\theta = 0$, then the sliding constraint reduces to:

$$\begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\theta}\end{bmatrix} = 0$$

Steered Standard Wheel

54

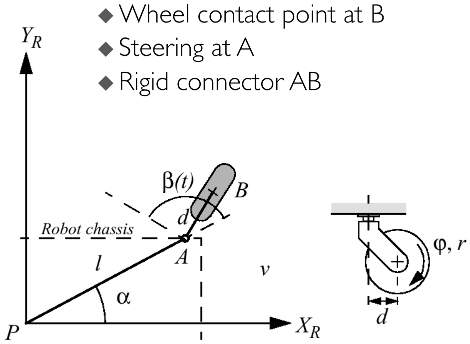
- This has all been for *fixed wheels*.
- For steered standard (spinning) wheels:
  - Same as fixed wheel, but β changes over time.
  - Instantaneously, it is fixed.



Castor (Offset) Wheel

55

- Wheel contact point at B
- Steering at A
- Rigid connector AB



Not Omnidirectional: Why?

56

$$\left[\sin(\alpha+\beta)\ \ -\cos(\alpha+\beta)\ \ (-l)\cos\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$
$$\left[\cos(\alpha+\beta)\ \ \sin(\alpha+\beta)\ \ l\sin\beta\right] R(\theta)\dot{\xi}_I - r\dot{\varphi} = 0$$

- Can constraints be satisfied for ANY $\dot{\xi}_I$?
- How will constraints be used?
- Once again, maneuverability / capability is…?

Inversely proportional to complexity of control

$$\textit{Capability} \propto \frac{1}{\textit{Control Complexity}}$$