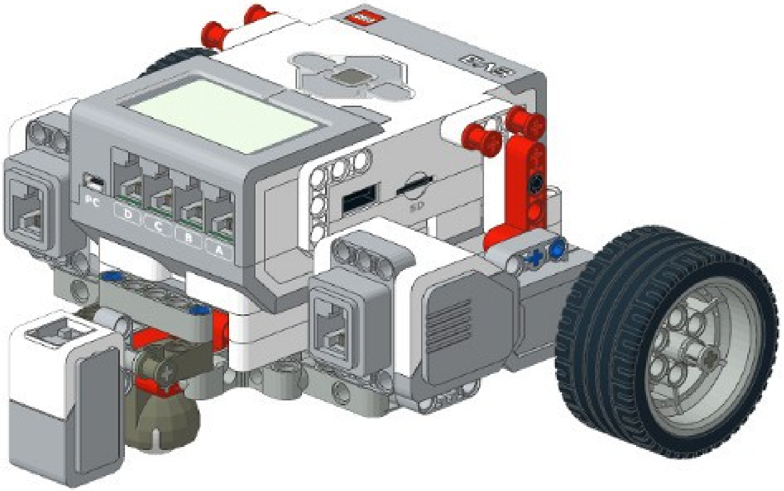
LEGO MINDSTORMS EV3

Gyro Turn



A POSTERIORI

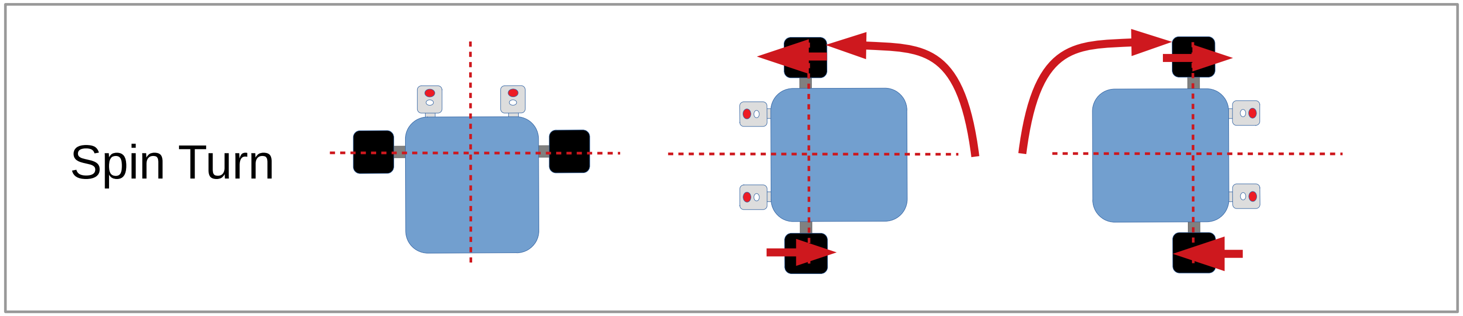
Play · Experience · Learn

Turning the Robot

- 4 Types:
  - Spin Turn
  - Left Pivot
  - Right Pivot
  - Curve Turn
- Differs in the location of the center of rotation

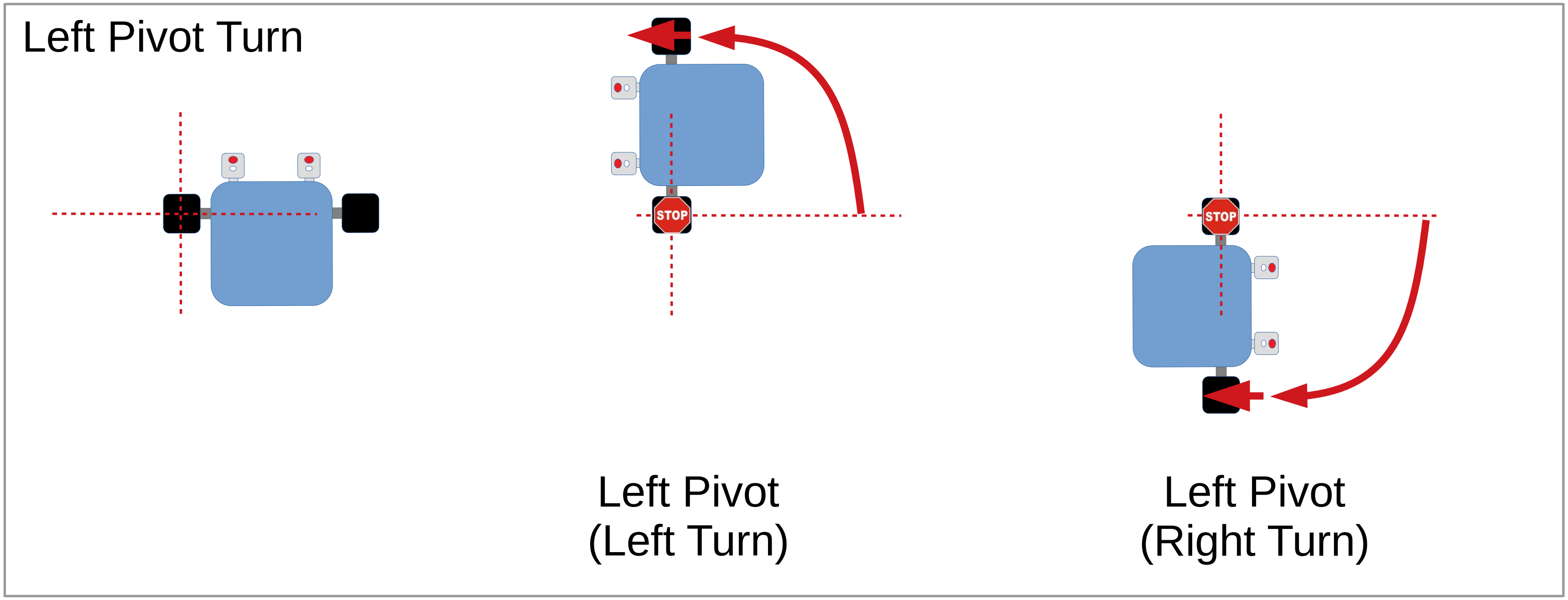
Turning the Robot

- Spin Turn
  - Both wheels turns at equal speed, but in opposite directions
  - Center of rotation is in the middle between two wheels



Turning the Robot

- Left Pivot
  - Left wheel stationary
  - Only right wheel moves
  - Center of rotation is on the left wheel



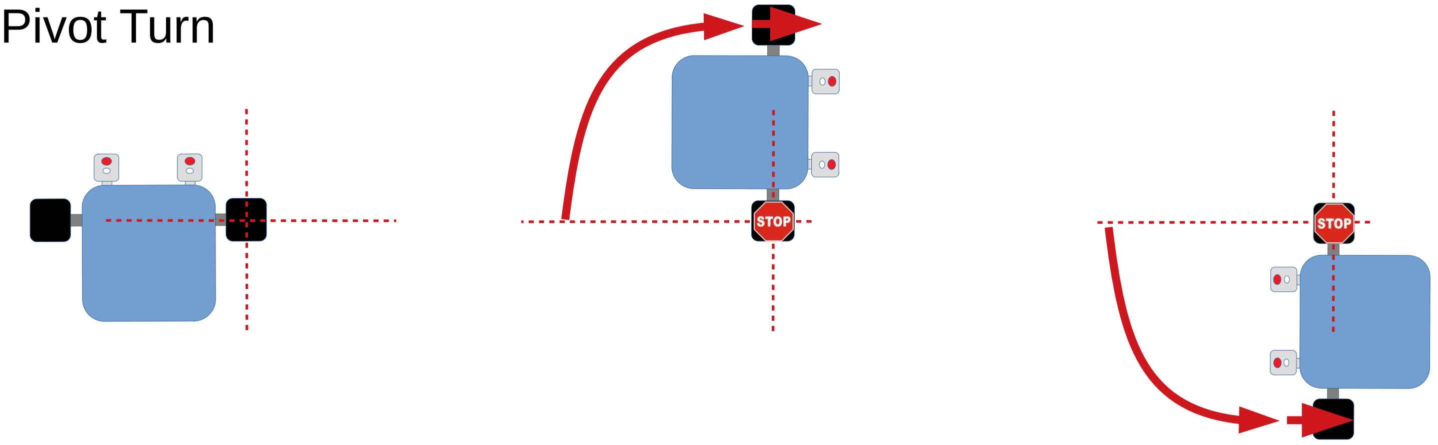
Left Pivot
(Left Turn)

Left Pivot
(Right Turn)

Turning the Robot

- Right Pivot
  - Right wheel stationary
  - Only left wheel moves
  - Center of rotation is on the right wheel

Right Pivot Turn



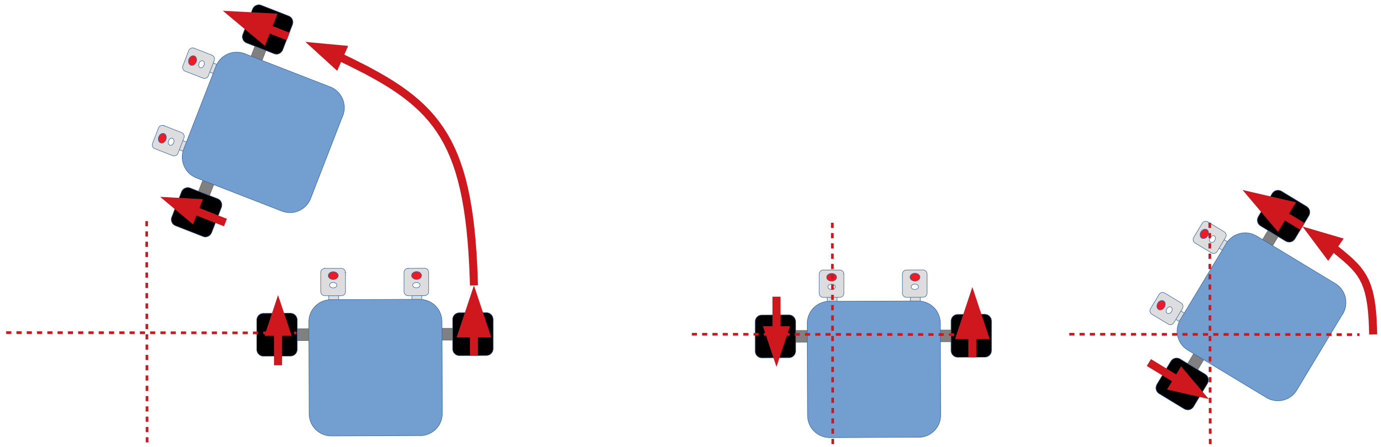
Right Pivot
(Right Turn)

Right Pivot
(Left Turn)

Turning the Robot

- Curve Turn
  - Both wheels moves, but at different speed
  - Center of rotation may be between the wheels or outside of the wheels

Curve Turn



Center of rotation outside of wheels
(Wheels same direction,
different speed)

Center of rotation between wheels
(Wheels different direction,
different speed)

Example Commands

Turns	Move Steering	Move Tank
Spin Turn	Steer: -100, Speed: 20 (left turn) Steer: 100, Speed: 20 (right turn)	Left: -20, Right: 20 (left turn) Left: 20, Right: -20 (right turn)
Left Pivot	Steer: -50, Speed: 20 (left turn) Steer: -50, Speed: -20 (right turn)	Left: 0, Right: 20 (left turn) Left: 0, Right: -20 (right turn)
Right Pivot	Steer: 50, Speed: 20 (right turn) Steer: 50, Speed: -20 (left turn)	Left: 20, Right: 0 (right turn) Left: -20, Right: 0 (left turn)
Curve	Steer: -100 to 100 Speed: Any	Left: Any, Right: Any

Curve Turn

- Most flexible
  - By setting the steering or left/right speed, a curve turn can replicate any of the turns (ie. spin, left pivot, right pivot)
- Hard to predict
  - Easy to predict how spin and pivot turns will behave
  - Hard to predict curve turns, as the center of rotation is not easily found

Gyro Turn

- Why?
  - Improve consistency of turns
  - Small errors in turns can result in large error in position
  - Especially important after line following, as the direction of the robot after line following is somewhat unpredictable

Basic Gyro Turn

1) Start Turning

2) Wait until robot reaches target angle

3) Stop

```
def gyro_spin_right(target_angle):
  move_steering(100, 20)
  while gyro_angle < target_angle:
    Pass
  stop()
```

Pseudo Code
Don’t copy it blindly; it won’t work
Read it, understand it, write your own

Basic Gyro Turn

- Problem:
  - Overshoot target angle
- Cause:
  - Robot can’t stop instantly. Momentum will carry it slightly pass the target angle.
- Solutions (3 options):

  1) Go slower

  2) Stop slightly before target angle

  3) Go fast, but slow down before reaching target angle

Basic Gyro Turn

- Solutions 1 and 2:
  - A combination of “Go slower” and “Stop slightly before target angle” can be adequate
  - Slow speed will waste some time
  - If that works well for you, go ahead!
  - Turns need not be perfect, as the gyro move will compensate for small errors
  - Recommend to keep error within 1 to 2 degrees

Intermediate Gyro Turn

- Solution 3
  - Go fast, but slow down before reaching target angle
  - Can apply any of the feedback control techniques used in line / gyro follower (eg. 2-states, 3-states proportional)
  - Use the feedback (gyro angle) to calculate the error (difference between gyro angle and target angle)
  - Use the error to control the turning speed

Intermediate Gyro Turn

- 2-States control
  - 1st State: error > 10 degrees; go fast
  - 2nd State: error < 10 degrees; go slow
  - Break: error is zero or less; exit loop and stop

```
def gyro_spin_right(target_angle):
  while True:
    error = target_angle - gyro_angle
    if error > 10:
      move_steering(100, 50)
    elif error > 0:
      move_steering(100, 20)
    else:
      break
  stop()
```

Pseudo Code
Don’t copy it blindly; it won’t work
Read it, understand it, write your own

Intermediate Gyro Turn

- Proportional control
  - Speed is proportional to error
  - Break: error is zero or less; exit loop and stop
  - Tune your gain accordingly

```
def gyro_spin_right(target_angle):
  while True:
    error = target_angle - gyro_angle
    correction = gain * error
    if error > 0:
      move_steering(100, correction)
    else:
      break
  stop()
```

<u>Pseudo Code</u>
Don’t copy it blindly; it won’t work
Read it, understand it, write your own

Intermediate Gyro Turn

- Tips
  - Error may be positive or negative depending on the direction of turn; adjust your code accordingly
  - Depending on your programming platform, you may get an error if the speed is too high
  - If the speed is too low, your robot may stop moving entirely
  - If using proportional control, you may want to constraint your speed to a minimum and maximum value to avoid these problems

Advanced Gyro Turn?

- Sudden acceleration may cause wheels to skid
  - Not an issue for pivot turns
  - Can change center of rotation for spin and curve turns
- Avoid this by increasing speed gradually
- Utility depends on robot’s weight, tire friction, motor power, etc. Some robots are more prone to skidding than others

Functions / My Blocks

- These are useful functions / my blocks that you should prepare
- Simple
  - Spin_Left (target_angle)
  - Spin_Right (target_angle)
  - Left_Pivot_Left (target_angle)
  - Left_Pivot_Right (target_angle)
  - Right_Pivot_Left (target_angle)
  - Right_Pivot_Right (target_angle)

Functions / My Blocks

- Intermediate
  - Spin (target_angle)
  - Left_Pivot (target_angle)
  - Right_Pivot (target_angle)
- Fewer functions / my blocks
- Figures out which way to turn by comparing current gyro angle with the target angle

Functions / My Blocks

- Advanced
  - Curve (target_angle, steering)
- Only one curve turn
- Figures out which way to turn by comparing current gyro angle with the target angle
- Steering parameter controls the center of rotation

Copyright

A POSTERIORI

Play · Experience · Learn