Control System Training

Module 11 – Position Control

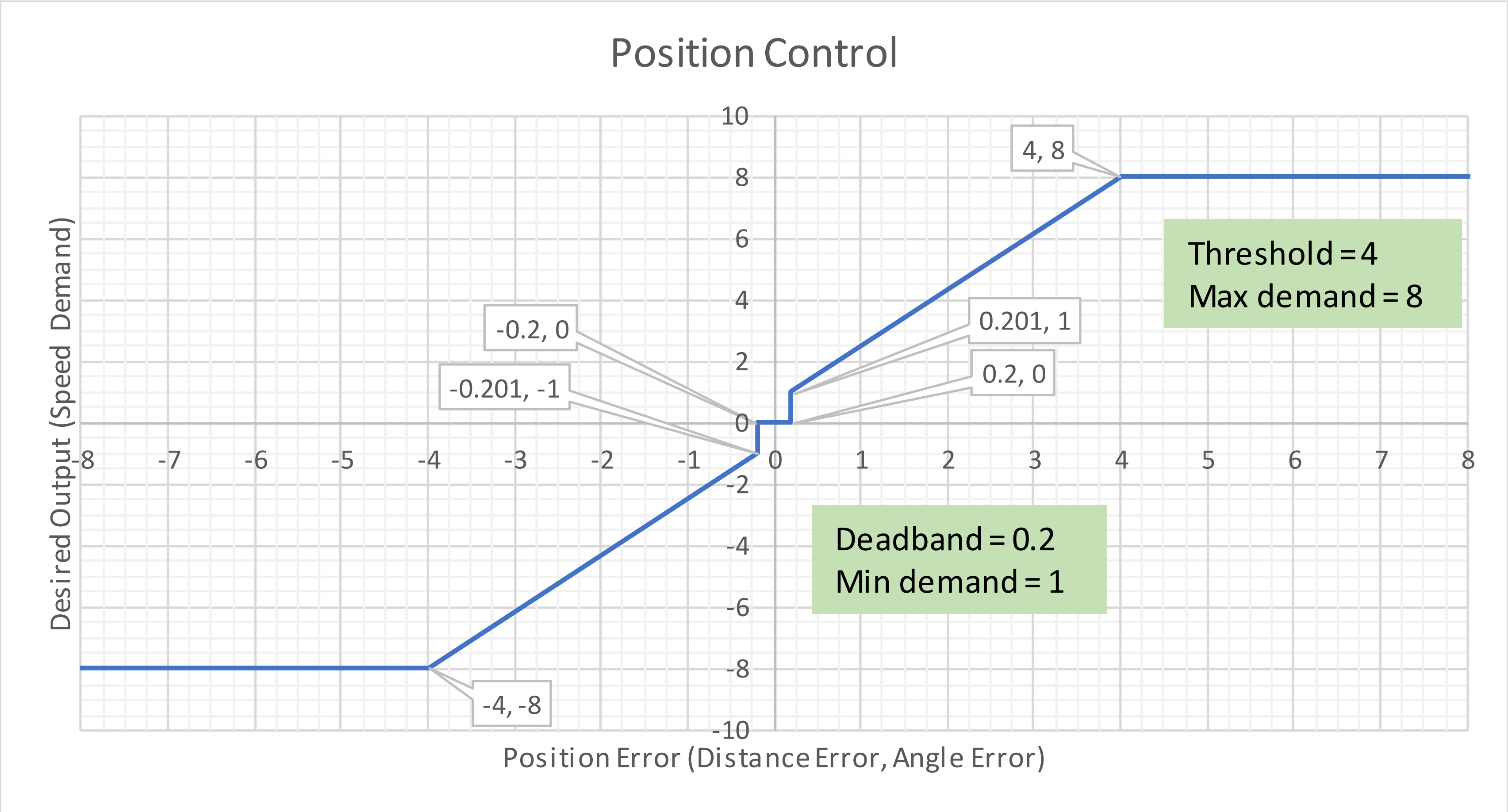
Copyright Notice

Position Control Examples

- **Spin robot X degrees**
  - Target angle = starting angle + amount to turn.
  - Turn until Current Angle = Starting Angle
  - Angle measured using Gyro
- **Move robot forward X feet**
  - Target position = starting position + amount to move.
  - Move forward until Current Position = Target Position
  - Position measured by drive system encoders
- **These are the SAME problem. Consider:**
  - Start at rest
  - End at rest
  - Slow down when getting closer to target
- **YES, there are more sophisticated solutions…**

Position Control Algorithm

- **Input – Error (angle error, distance error)**
- **Output – Wheel Speed Demand**



Position Control Algorithm

- **Input variables**
  - Target position
  - Current position
    - Error is calculated from these
- **Input parameters**
  - Largest output
  - Smallest output
  - Error threshold
    - Where largest output starts to ramp to smallest output
  - Error deadband
    - How close is considered "on target"
    - Where output drops from smallest output to zero
  - Consecutive times within deadband to be considered done
    - Accounts for sliding through target

Position Control - Tuning

- **Max Speed**
  - Don’t make it faster than your robot can handle
  - If this is too slow, it will take a long time to get to the target
- **Min Speed**
  - If this is too fast, the robot will overshoot the target and oscillate
  - If this is too slow, the robot may stop, never reaching the target, if it can’t control at the slow speed.

Position Control - Tuning

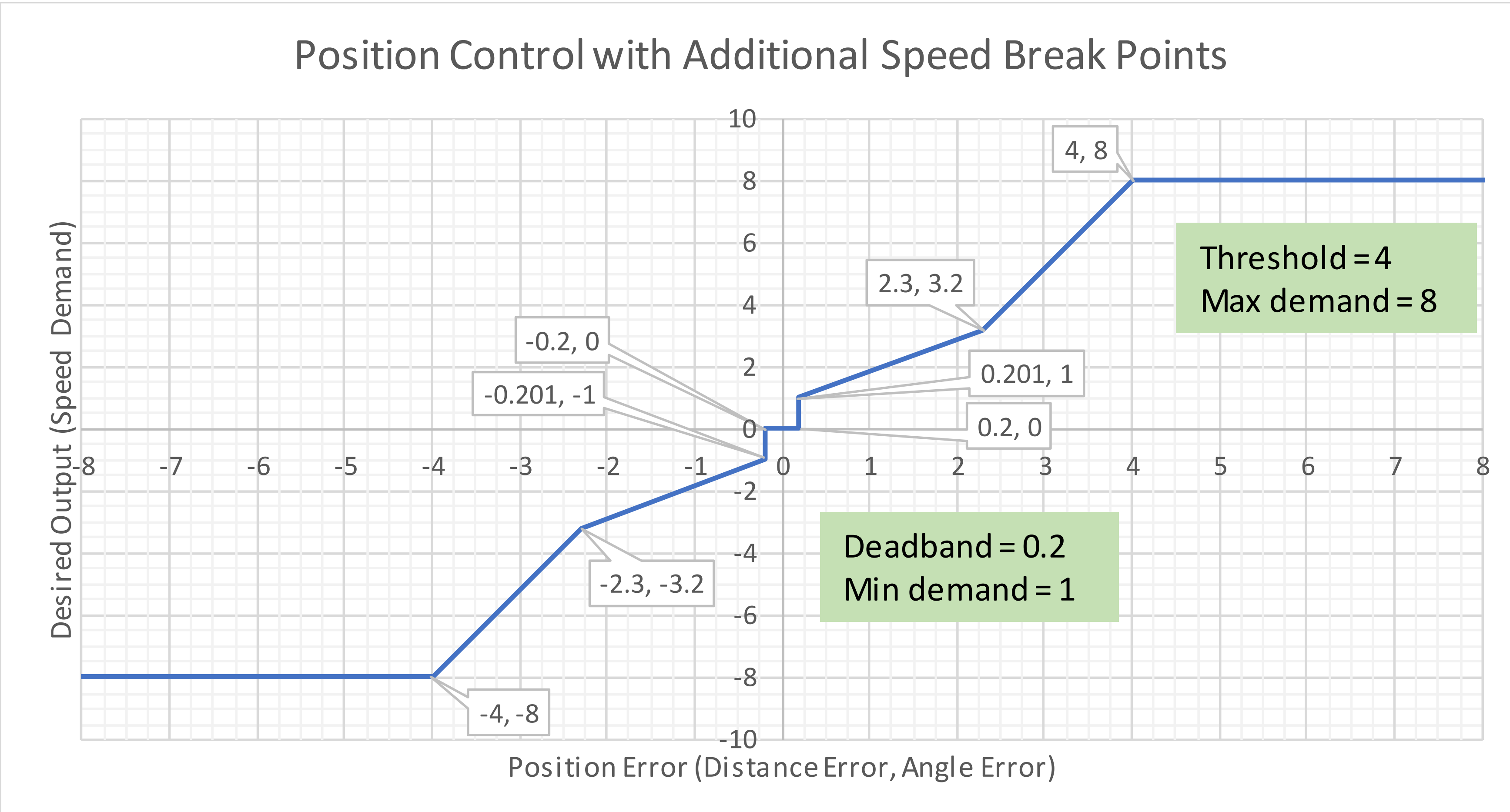
- **Threshold**
  - If this is too far away, the action will take longer than needed
  - If this is too close, the robot's momentum will cause the speed to be greater than desired
- **Deadband**
  - Set this to get the accuracy you need
  - If this is too large, accuracy will suffer
  - If this is too small, robot may oscillate around target.

Position Control – Setting the Parameters

- **Set initial parameters based on how your robot performs**
- **Test and tune parameters to balance speed and accuracy.**
  - This will require time and a working robot!
- **The same tuning should work for all similar movements**
  - One set of tuning for moving forward / backwards
  - One set of tuning for spinning movements
- **For any autonomous action also include a timeout!!!**
  - If you measured the distance wrong and hit the wall, the robot needs to continue…

Alternate Position Control Algorithm

- **Alternate version with additional error, speed demand break point pairs**



Exercise 11.1

- **Write the algebraic expressions for the position control algorithm output when:**
  - Error >= threshold
  - Error <= -threshold
  - Absolute value of error < deadband
  - Error >= deadband and error < threshold
  - Error <= -deadband and error > -threshold
- **The output should be a function of:**
  - Maximum speed
  - Minimum speed
  - Zero
- **Question: What controls the speed from 0 to the maximum speed when the position control is first started?**
- **Question: What analog algorithm could be used to implement this?**

Exercise 11.2

- **Write a generic state machine to spin the robot.**
- **The inputs are:**
  - Maximum speed (Ft/Sec)
  - Amount to spin (Degrees) (+ = clockwise, - = counterclockwise)
  - Timeout time (seconds)
  - Consecutive
- **Outputs (while in progress)**
  - Wheel speed demand (FT/SEC)
- **Outputs (when completed)**
  - On target digital (If not on target, then timeout occurred.)
- **Hint: This state machine only has 3 states.**