Linear Motion Profiles

2015 – SCRIW

Disclaimer & Credits…

Technical content was provided by FRC Teams 251 - The Cheesy Poofs, Team 971 - Spartan Robotics, and Paul Copioli, President Vex Robotics.

Resources

- 254 and 971 Presentation – 2015 World Championship
  - http://www.chiefdelphi.com/forums/showthread.php?t=136798 (announcement)
  - https://docs.google.com/presentation...e&delayms=3000 (presentation)
  - https://youtu.be/8319J1BEHwM (presentation recording)
- Paul Copioli – Chief Delphi Post
  - http://www.chiefdelphi.com/forums/showthread.php?t=98358 (discussion thread)
  - http://www.chiefdelphi.com/forums/showpost.php?p=1204107&postcount=18 (Paul's post)

Linear Motion Profiles: Topics

- Got to move? You have options…
- How to use a profile
- How to plan a profile
- How to calculate a profile

- Focus of discussion:
  - Profiles in autonomous routines
  - Linear (1 dimension profiles)

Hint: Motion profiles are cool

GOT TO MOVE? YOU HAVE OPTIONS…

Problem statement

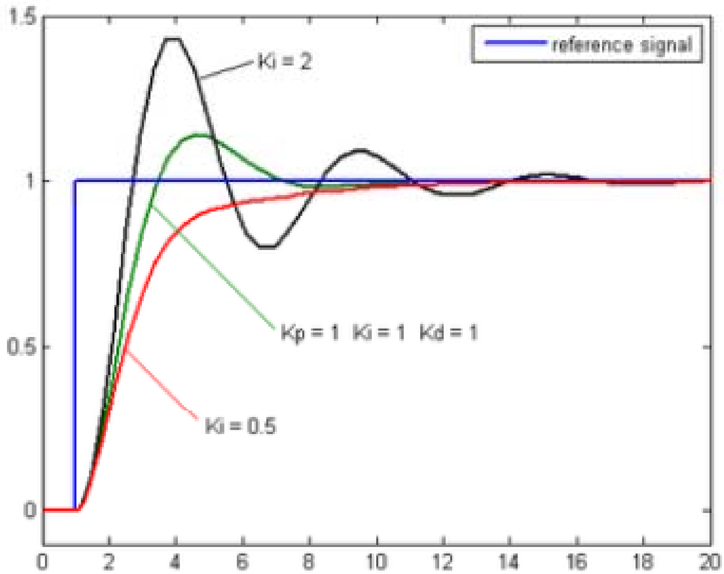
- We need to drive forward 10 feet in autonomous to get in scoring position
- We need to lift an elevator from 10” above the floor to 40” above the floor… then return to 10”
- We need to rotate an arm from 30° to 75°

Open loop – time based

- Plan 5 ft / sec
- Find motor power that runs at 5 ft / sec
- Turn on motors at that power for approximately 2 secs
- *Are we there yet?*

ISSUES?

"PID" Closed loop feedback control



$$u(t) = \overbrace{K_p e(t)}^{\text{Proportional}} + \overbrace{K_i \int_0^t e(\tau)\,d\tau}^{\text{Integral}} + \overbrace{K_d \frac{d}{dt} e(t)}^{\text{Derivative}}$$

Proportional	Integral	Derivative
Slow down near the goal	*Overcome friction*	*Add damping*

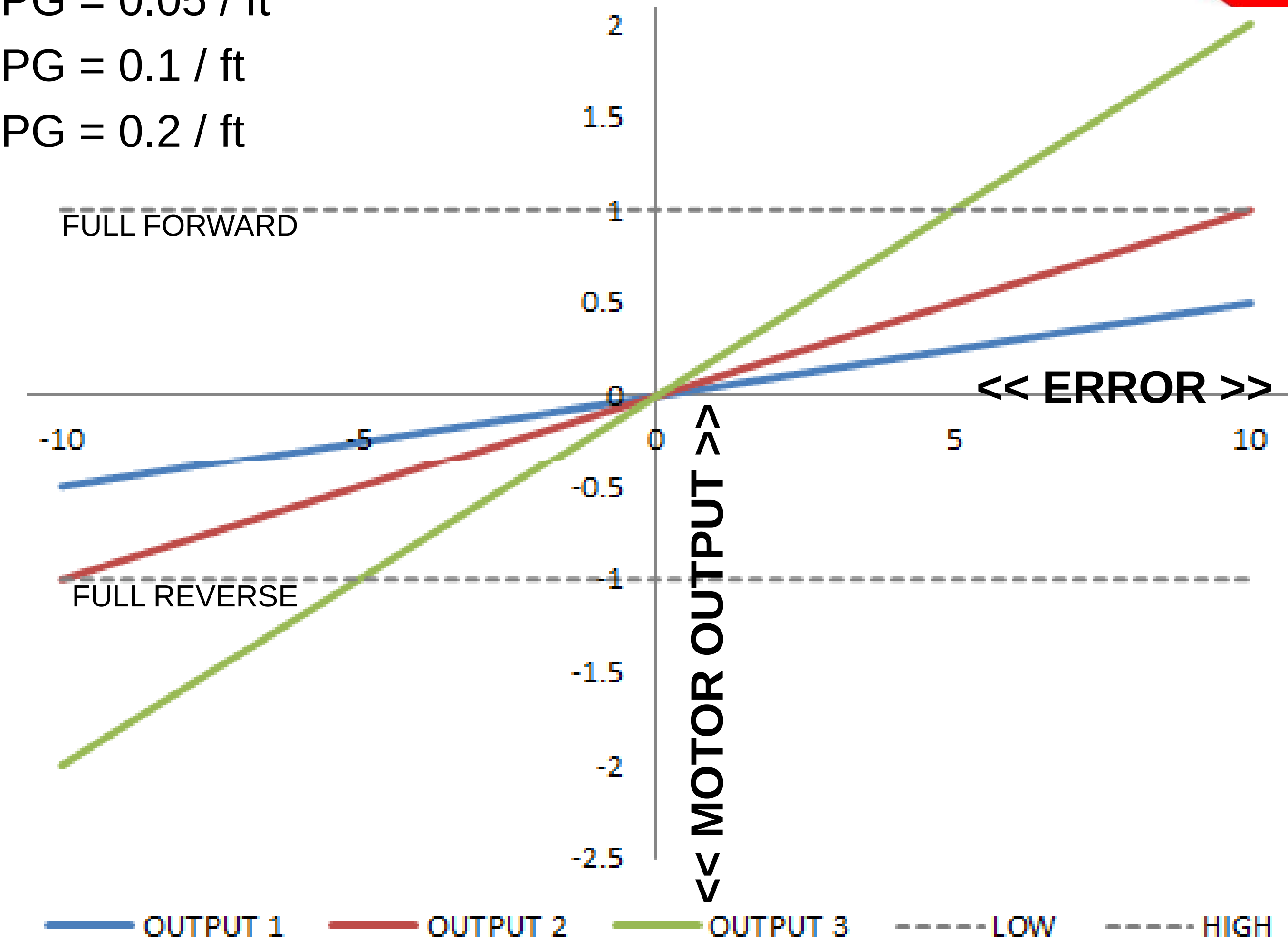
ERROR SIGNAL = TARGET – CURRENT DISTANCE

- Requires drive train encoder for distance feedback
- Also *recommend* gyro scope for maintaining heading

“PID” Closed loop feedback control

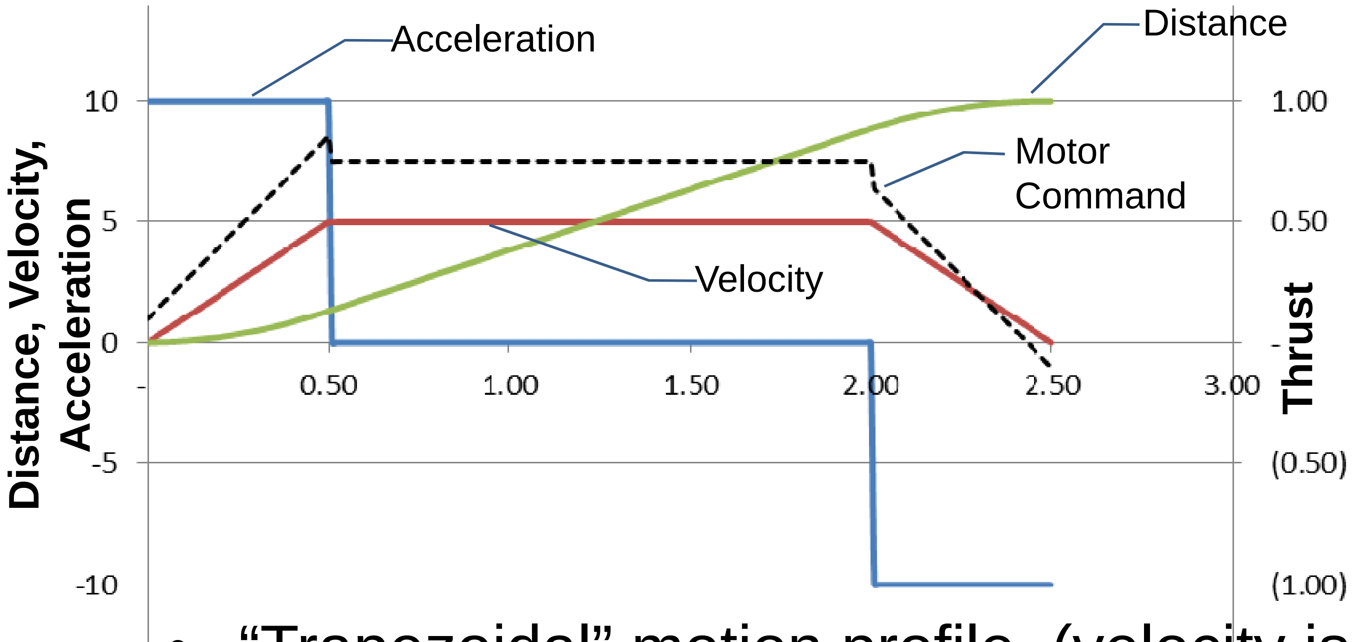
Motor output (PG only)

- Output 1 PG = 0.05 / ft
- Output 2 PG = 0.1 / ft
- Output 3 PG = 0.2 / ft



Linear Motion Profile

Feed forward + feedback control



- "Trapezoidal" motion profile (velocity is trapezoid)
- 3 Phases of motion:
  – Constant acceleration
  – Constant speed ("cruise speed")
  – Constant deceleration (mirror of acceleration phase)

Linear Motion Profile

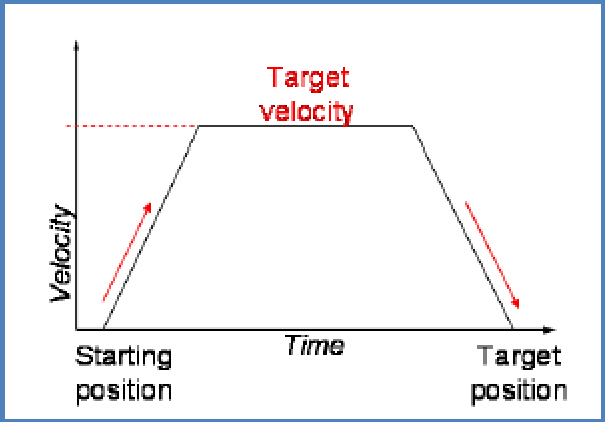
- Profile calculated from motion kinematics equations

$$v_f = v_i + a\,t$$
$$x_f = x_i + v\,t + ½\,a\,t^2$$
$$x_f = x_i + t\,(v_i + v_f)\,/\,2$$

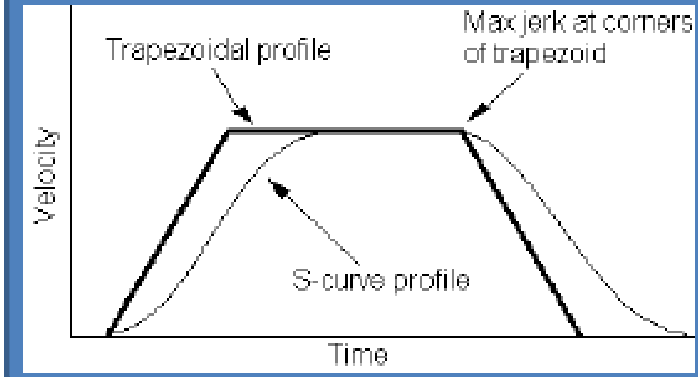
- Profile method advantages:
  - Smoother acceleration / deceleration than PID method
  - Easier to plan routine that fits within robot speed / acceleration constraints
  - ***<u>MUCH</u>*** easier to "tune" for repeatable performance

Linear motion profiles

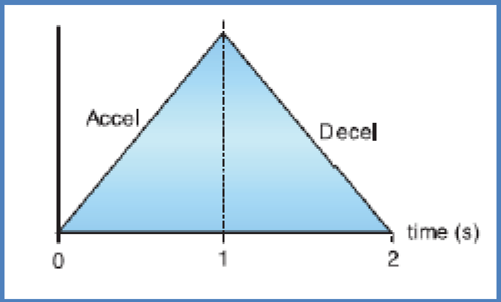
- 3 velocity profiles used:



TRAPEZOID



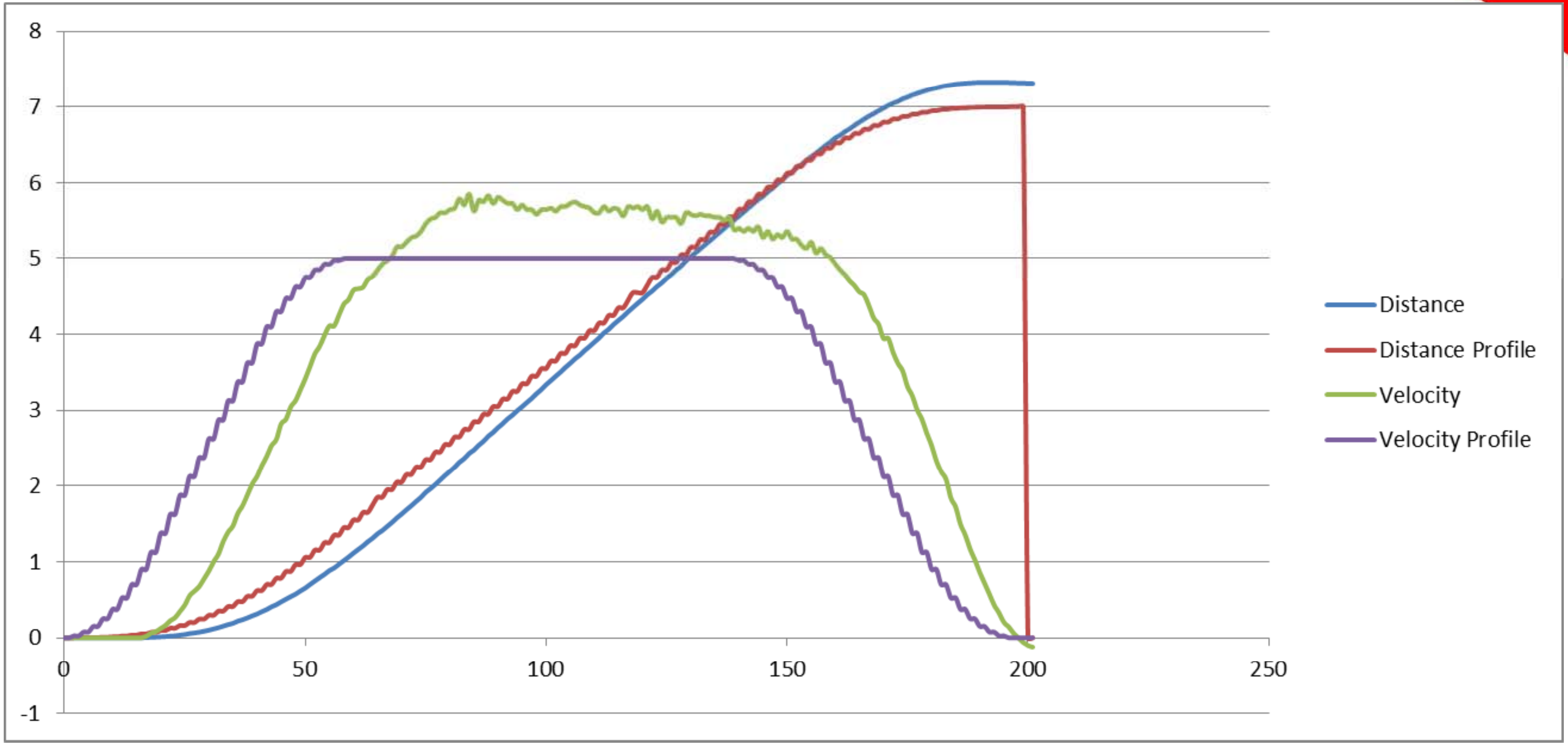
"S"-CURVE



TRIANGULAR

S-Curve profile test

Profile with sensor feedback readings



Simplified case: Autonomous mode

HOW TO USE A LINEAR PROFILE

Basic steps

Autonomous mode profile

1. Profile is stored in an array**
   - Columns: Time, Velocity at time “x”, Distance at time “x”, Acceleration at time “x”
   - Array is calculated at start-up, or is read from a spreadsheet
2. “Playback” profile in a timed loop
3. At each loop iteration, calculate motor power command from the profile

Power Out = Profile Velocity * Velocity Gain **+**

Profile Acceleration * Acceleration Gain **+**

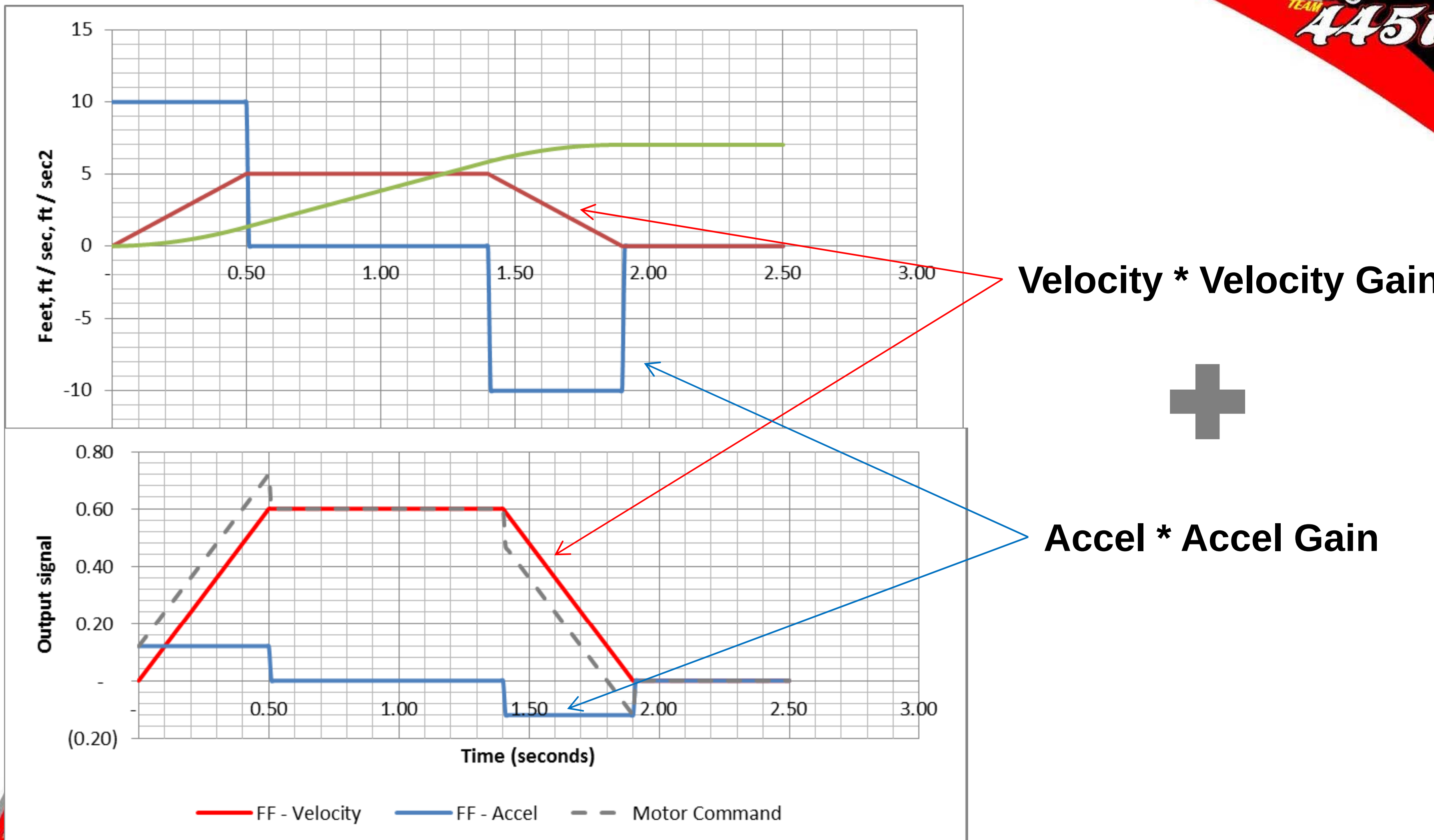
Feed forward

(Profile Distance – Actual Distance) * PID Gains

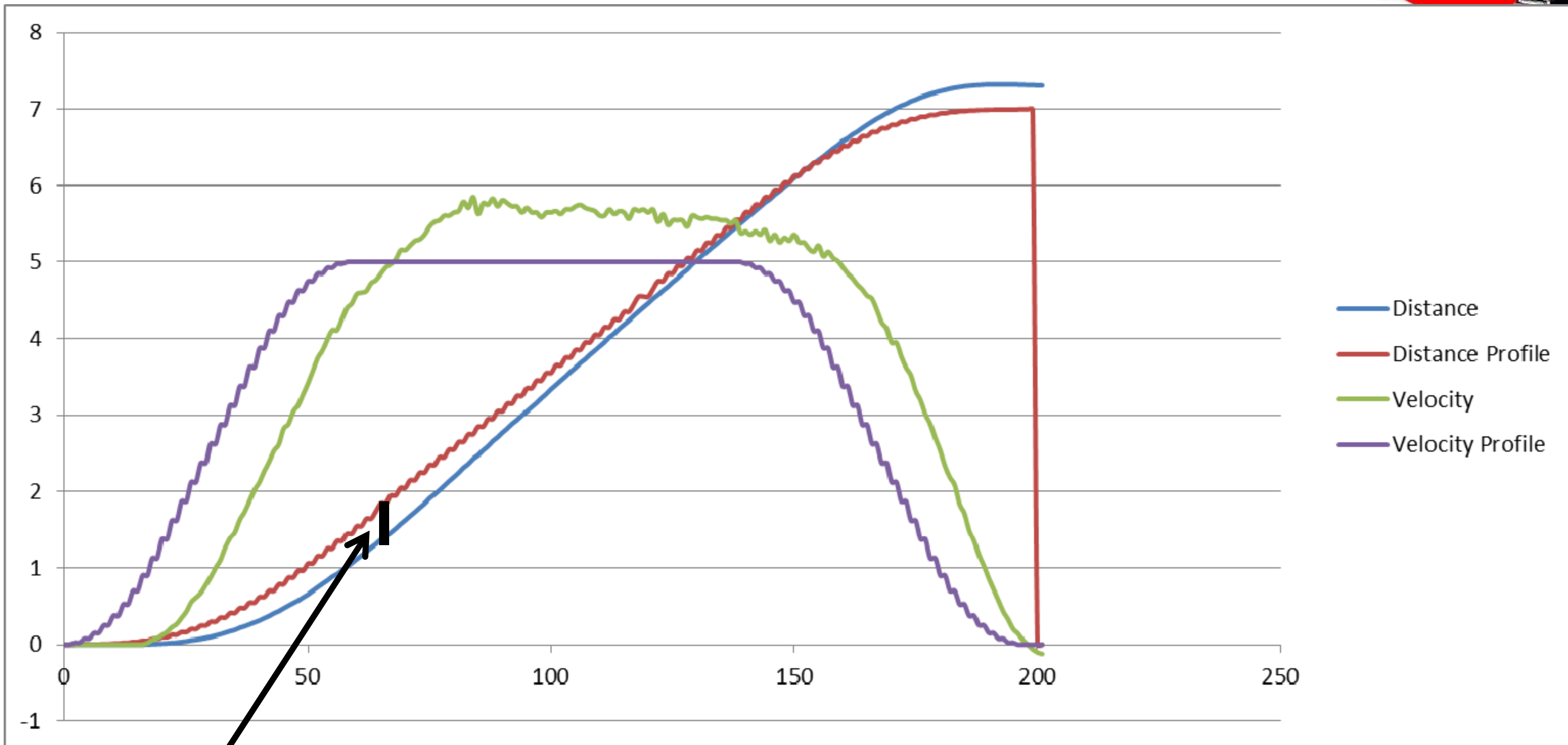
Feed back

*** Note: Profile can be calculated dynamically as well.*

Feed forward control



Feedback control



Gap between actual distance and profile distance = “error” signal

<u>Output Signal</u> = Feed forward values **+** error * PID gains

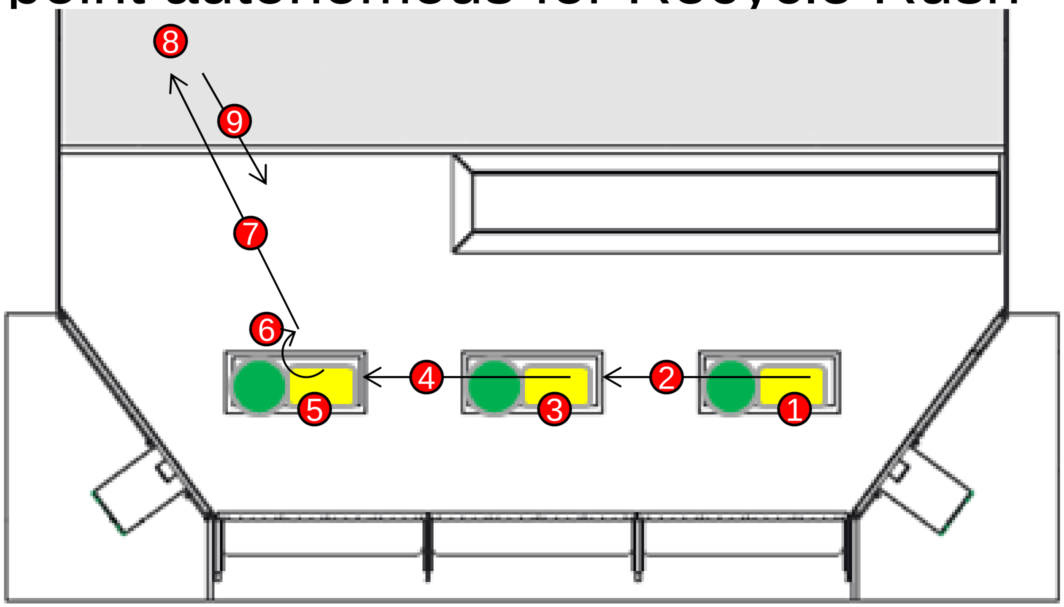
Simplified case: Start and end at 0 velocity

HOW TO PLAN A PROFILE

Plan your move

- Layout distance / heading plan
  - Can you accomplish your goal with one move?
  - Do you need to turn?

20 point autonomous for Recycle Rush



Steps

1. Pick up tote
2. Move forward (profile 1)
3. Pick up second tote
4. Move forward (profile 2)
5. Pick up 3rd tote
6. Turn
7. Move forward (profile 3)
8. Drop tote stack
9. Move backward (profile 4)

Know your constraints

- AVOID!
  - Accelerating drive train too fast → *wheel slip…*
  - Accelerating / decelerating too fast → *MOMENTUM*
  - Cruising too fast → *can't control profile near full throttle*
- Mechanical issues cannot be fixed by software!
  - Motor / gearing needs to be sized for the task
  - High friction / binding can cause unpredictable results
- Use good sensors
  - High quality encoders, potentiometers, and gyros are critical to success

Plan your code sequence, test, and tune

- Take your autonomous plan and generate pseudo-code
  - Initialize, calculate and/or read in profile, run profile, next step…, end
- Set the following profile constants. *For example…*
  - Profile loop frequency = 100Hz = 10 ms loop time
  - Cruise velocity = 5 ft / sec
  - Acceleration = 15 ft / sec^2
  - Target distance = 7 feet
- Test profile
  - Tune with velocity gain only – get close to correct cruise speed
  - Add acceleration gain – get close to correct distance
  - Add PID gain for distance correction

Other considerations

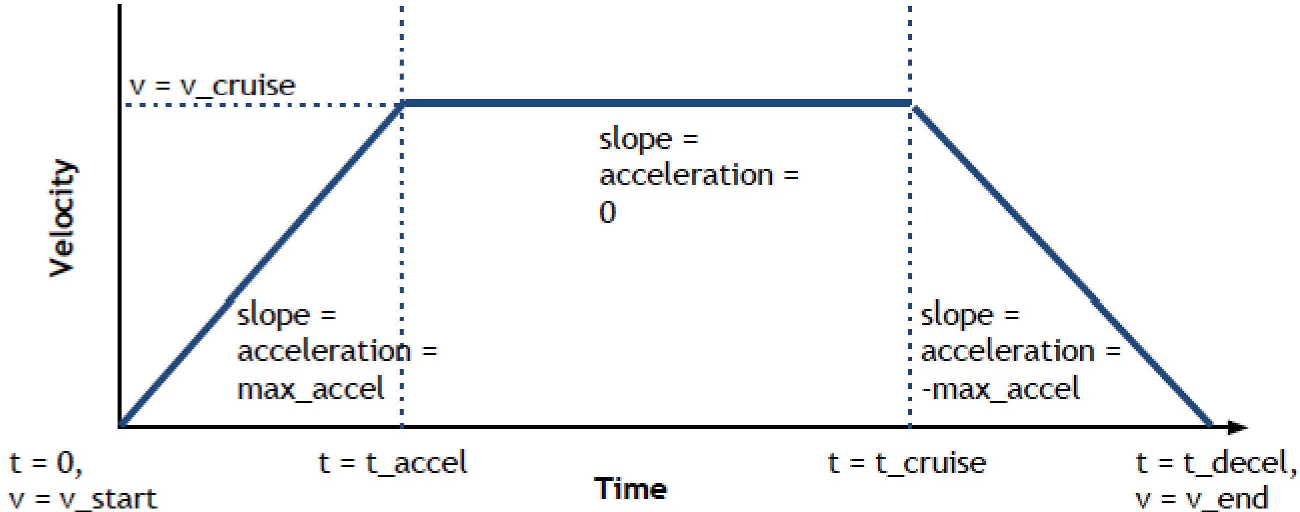
- Check tuning under different loads
  - Lifting a load vs. lowering a load
  - Moving with a game piece vs. without
  - Drive train pushing an object vs. no object in way

- Check tuning under different battery conditions

Boxcar filter and polynomial methods

HOW TO CALCULATE A PROFILE

Trapezoidal Profile



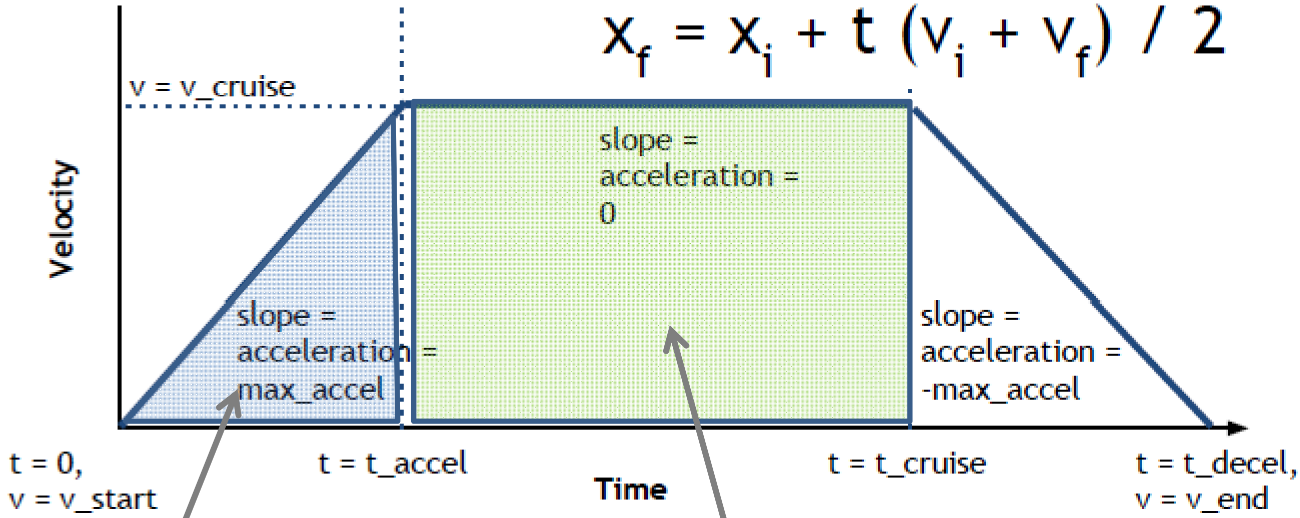
- Profile is symmetric – acceleration / deceleration rate and time are equal
- Many solutions to the same distance…
  - Set max acceleration and cruise velocity based on system performance
  - Pick your target distance

Trapezoidal Profile

$$v_f = v_i + a\,t$$

$$x_f = x_i + v\,t + \tfrac{1}{2}\,a\,t^2$$

$$x_f = x_i + t\,(v_i + v_f)\,/\,2$$



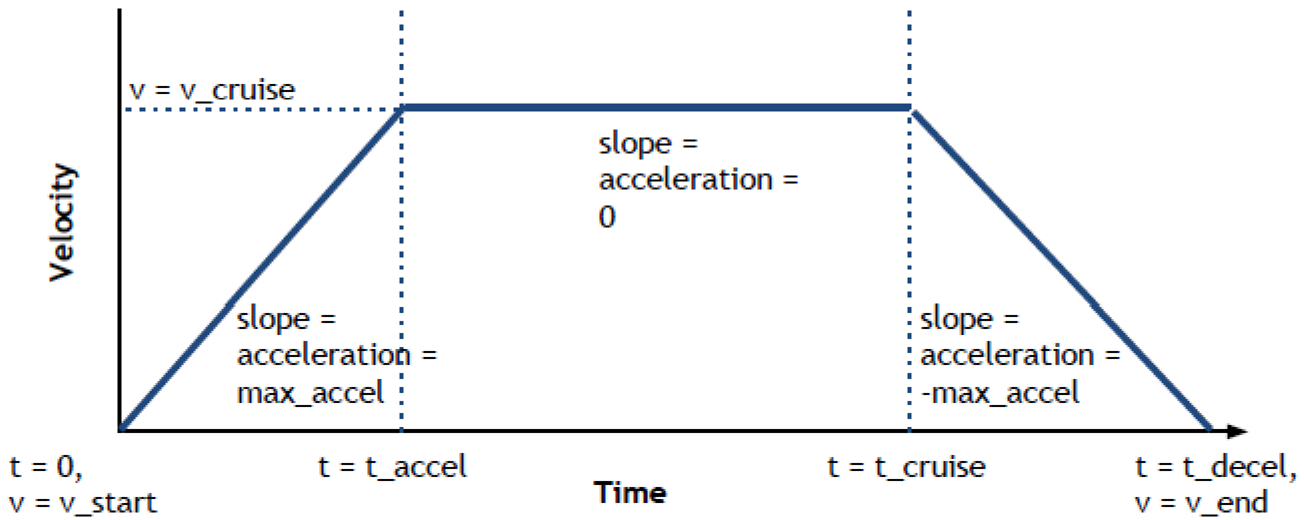
Distance = ½ * accel * t_accel²

-- or –

Distance = ½ * v_cruise * t_accel

Distance = v_cruise * (t_cruise – t_accel)

Trapezoidal Profile



Target distance = v_cruise / (t_accel + t_cruise)

Trapezoidal Profile Calculation Techniques

- Polynomial method
  (technique described by 254 / 971)

- Boxcar filter method
  (technique described by Paul Copioli)