Improving Robot Reliability

Written in response to questions from another FLL Team

Sources of Problems

| Problem	Impact
Alignment in base varies from run to run	Each run is different and missions sometimes work.
Robots don't travel straight for long or turn exactly the same amount	It is hard to predict the robot location exactly.
Errors accumulate as you travel	Long missions tend to fail. It is hard to do missions far from base
Adjusting motors/attachments in base	First move out of base may behave differently each time. Attachments don't work the same each time
Battery levels impact motor performance	Tweaks that work today fail tomorrow

Starting Points in Base are Critical

- FLL teams need to figure out where to start in base
  - **Jigs:** a LEGO ruler/wall that your robot can align against them in base
  - **Same start each time:** pick one spot and start there no matter what the mission for easy starts
  - **Inch marks:** Use the inch marks to pick a starting spot for each run
  - **Words:** Base has words. If you aren't near an inch mark, pick a word or letter to start on.

- Even better, try to find a way to align the robot using other techniques (see next page)

Use a jig

Use marks
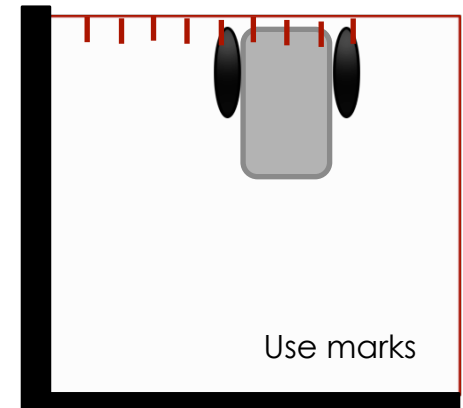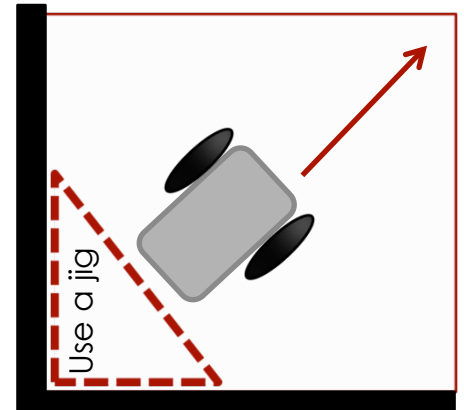
# Robot Doesn't Travel Straight & Errors Accumulate Over Time

- By the time you get to the far side of the table, you are no longer in the right position

- Solution: Repeat alignment techniques multiple times in a run for better reliability (see next slide)

Mission Model 1

Mission Model 2
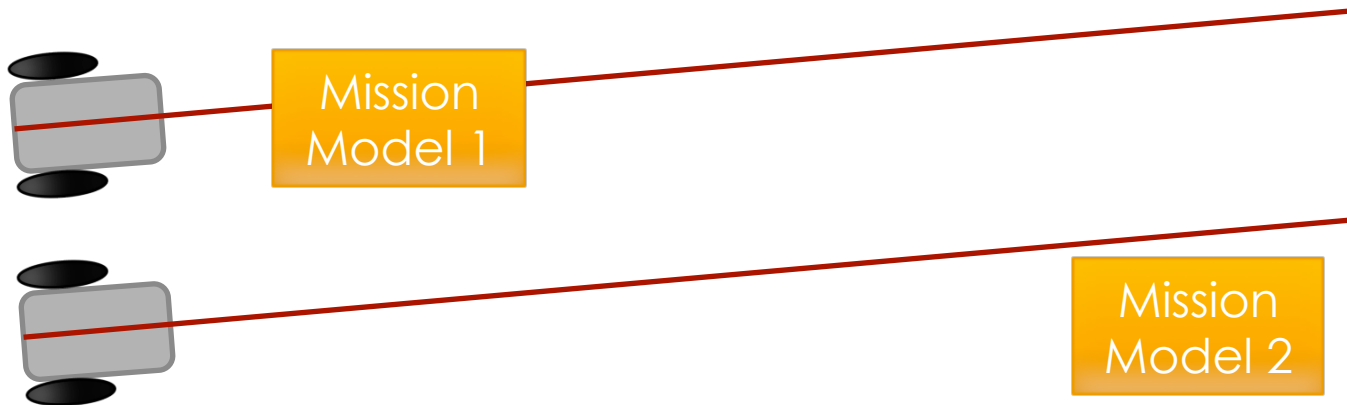
Where are you on the FLL table?

- Consider these alignment strategies that are commonly used:
  - **Align on walls** – deliberately back into a wall to straighten out (note: You may stall doing this. See the Stall Detection Lesson)
  - **Square/Align on lines** –If you are moving angled, you can straighten out whenever you see a line. (See Squaring Lesson)
  - **Move until a line** – travel until you find a line so you know where you are on the mat (See Basic Lessons)
  - **Align on a mission model** – Mission models that are stuck in one place can be used to align against

Back into walls

Square on a line

Align on a mission model

Mission Model
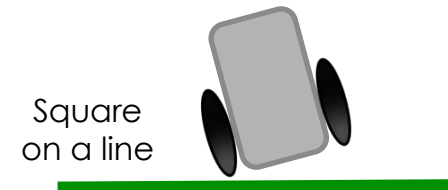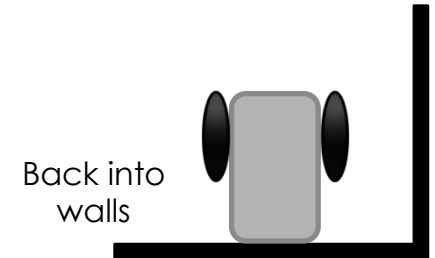
Adjusting Attachments in Base

- Just like the robot body, you need to set up your attachments in the same way each time for improving reliability
  - Jigs that allow the attachment arm to only move to a certain level to make sure the arm is set the same way each time
    - In Senior Solutions, we used a jig to make sure the arm that picked up the pill box always started at the right level
  - Indicators on the robot (e.g. bright peg) might help you remember where to reset the arm to
    - In Food Factor, we had a red peg in a hole to remember how far back to move the arm
  - You can use a touch sensor to detect the position of an attachment at the start of a run

Adjusting Motors in Base

- Moving attachments or wheels
  - When the program is stopped you can move wheels and attachments easily and it has no impact
  - If a program is running, there are multiple steps
    1. You need to put the motors into "coast" mode
    2. If you move the motors in coast mode, the motors will move back to the original position on the first move!
       - You need to "reset" the motor after an adjustment and before you start your run

1) Put all the motors you use on coast so you can move the motors by hand to adjust
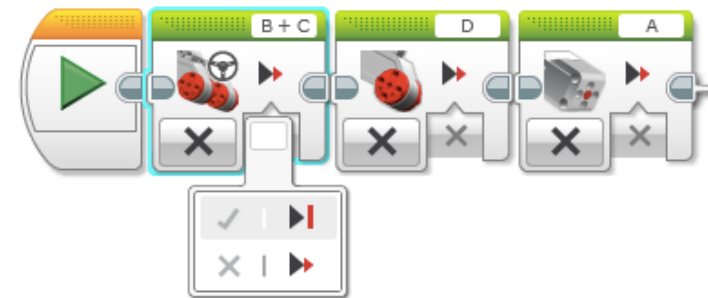
2) Now you have to "reset" the motors
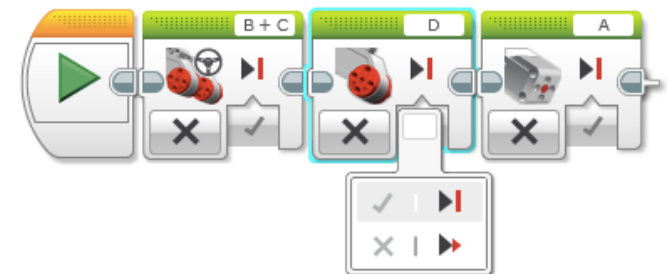
Using Coast

This set puts all the motors in coast mode. You should be able to freely move your motors by hand without any resistance.

When you are ready to start your mission, hit the middle button

We tell motor Arm "A" to move 10 degrees. It will move 10 degrees from where it was when the program last started and not where you moved it to by hand earlier

This code shows that the motor arm (A) will not be predictable no matter how much you reset the arm by hand. It's movement is based on where the arm last was. Move the arm by hand at least 90 degrees to see the difference.

Doesn't work well. Not as reliable!

Using Coast & Reset

This set puts all the motors in coast mode. You should be able to freely move your motors by hand without any resistance.

When you are ready to start your mission, hit the middle button

Here, we added a "reset" step.

Now when you ask the attachment arm "A" to move 10 degrees, it will move 10 degrees from where you manually moved the arm when it was in base.

This code shows that the motor arm (A) will be more predictable because you are able to set a starting position for the arm in base. Move the arm by hand at least 90 degrees to see the difference.

More reliable!

Other Factors in Reliability

- **Battery life**
  - If you program your robot when the battery life is low, it won't run the same when fully charged
    - Motors behave differently with low battery
    - But using sensors makes you not as dependent on battery

- **LEGO pieces come apart over time:**
  - Squeeze in LEGO pieces in key areas before a run – the pegs get loose which means the sensors may not be in the same place as a previous run
  - Push wires in for sensors and motors. They come out!

- **Motors and sensors don't always match:**
  - Some teams test motors, sensors and wheels to make sure that they match
  - You will never get a perfect match so we recommend use other techniques and accept that they will be different

Credits

- This lesson was written by Arvind Seshan and Sanjay Seshan from FLL Team Not the Droids You Are Looking For

- These lessons are free to use and adapt. Just give credit to the original authors of the lesson

- More lessons at: www.ev3lessons.com