ROBOT PERFORMANCE FUNDAMENTALS
KEYS TO ROBOT PERFORMANCE

• Fast, reliable robot
  – Goes straight
  – Robustly built. Won’t break apart in handling.
  – Easy to swap attachments
  – Easy to align

• Fast mission transitions

• Doing multiple things per mission

• Consistent launching from base
BUILDING A FAST, RELIABLE ROBOT

• Consider starting with the reference robot design found in the *Building a Competitive Robot* training module. This design evolved over six seasons of very successful table performance. It incorporates many aspects for fast transitions and robustness.

• These aspects will be discussed later in this training module.
GOING STRAIGHT

• KEY POINT: The single most important thing for the robot to do is to go straight.

• Why? Most of the time the robot will be “dead reckoning” and not using sensors or side wheels in its movement. A robot that doesn’t go straight will be far less likely to reach its desired location.

• Elements of a robot that goes straight:
  – Matched motors and wheels.
  – No rear third wheel that can swivel. Either use a skid or two fixed wheels without tires.
  – A very rigid body like a high performance sports car.
  – Equal traction on motored wheels.

• Check straightness by aligning robot to wall and having it go over straight line at various power levels. Measure deviations. Use test 50.rbt – test 100.rbt to check.
UNBALANCED ROBOT

- Swap wheels and motors to get the most consistent performance.
- If the robot still pulls left or right and gets progressively worse at higher power levels, the most likely cause is a mass imbalance over the two motored wheels.
- This can easily happen with an arm motor that is not directly over the center of the robot. The arm motor weighs 3 ounces and puts additional downward pressure over a wheel, giving it better traction. The other wheel with less traction will slip more as power increases.
UNBALANCED ROBOT

• Solution?
  – KEY POINT: Put a counterweight over the opposite wheel to balance arm motor. If you can, get hold of the 2 x 6 x 2 “heavy crane” brick from the 2006 Nano Quest challenge (part no. 73843). It weighs 2 ounces.
  – Move the arm motor closer to the front middle of the robot.
  – Don’t run the motors at maximum power.
  – Take into account how far the robot will pull to the left or right when routing robot.
IMPORTANCE OF CONSISTENCY

• Goal should be to get every mission to work at least 90% of the time.

• For example, let’s assume that the robot has six missions programmed. Probability that they all will work = (0.9)^6 = 53%. In other words, roughly 1 in 2 chance of a flawless round.

• It is very easy for team to stop further effort on a mission when it works for the first time.

• KEY POINT: A mission isn’t finished until it works at least 9 out of 10 times in a row!
IMPORTANCE OF CONSISTENCY (continued)

- KEY POINT: At best, performance on competition table will match practice table, but often will be worst!

- Over six years of competition, Sea Monsters have completed 48 rounds at the regional and championship tournaments. Perhaps four times have the Sea Monsters completed all of our programmed missions perfectly, despite doing so numerous times at our practice table. A couple of important conclusions:
  - It is really hard to be perfectly consistent in competition, mostly due to differences in table and conditions.
  - Sea Monsters have been very successful at regional and championship tournaments. Part of this was due to being more consistent than the competition, and part of this was due to fast transitions allowing more missions to be done and higher potential maximum scores to compensate for missed missions.
FASTER TRANSITIONS

- With only 150 seconds in a round, time is the enemy.
- Slow transitions are deadly.
  - At the 2011 VA/DC FLL Championship, the Sea Monster robot left base 9 times, averaging 5 seconds per transition. Total transition time = 8 transitions x 5 seconds = 40 seconds.
  - Typical teams take 10 seconds per transition, using up another 40 seconds.
WHY SLOW TRANSITIONS?

• Teams have N programs, one for each mission.
• At each transition, they have to push the arrow button to find the program, set the robot down, align the robot and then push the orange button.
• Issues:
  – Time consuming
  – In the stress of the tournament, it is easy to make errors
TIMED MASTER PROGRAM

• More advance teams combine the missions into a master program that switches between missions on a timer.

• Issues:
  – In the stress of the tournament, it is easy to take too long in swapping attachments, aligning robot, etc. This will blow the next mission.
  – To avoid this, the timer must have extra length, costing valuable seconds for each transition.
SUPERIOR SOLUTION: TOUCH SENSOR AND BLOCKS

• KEY POINT: Use the Sea Monster trick of turning each mission into a “block” (equivalent to a subroutine). Each block is triggered by pushing the touch sensor.

• Huge advantages:
  – One program, no searching between missions.
  – Team has time to swap attachments, align the robot, etc. to prepare for the next mission.
  – As team gets faster in transitions, every second is saved.
• Open *Template.rbt* and immediately File -> Save As with simple name. Warning: Don’t simply save, which will just update *Template.rbt*.

• Next, customize template for mission.
TEMPLATE EXPLAINED #1: UNLOCK MOTORS

- Start of each block begins where last block stopped.
- Should last step in previous mission brake a motor, the motor can’t be moved unless released, which this step does.
In the heat of competition, it is important to know which mission is next.

Simply type the mission name in the Simple Text box.

Center text in right display box by changing X or dragging text with mouse.

Text will be displayed at start of mission.
TEMPLATE EXPLAINED #3: TOUCH SENSOR

- This launches robot when team is ready.
- Bumped means pushing touch sensor in and then releasing.
- At beginning of competition, push sensor when “3, 2, 1” is announced and release at the “L” of “Lego”.

![Diagram of Touch Sensor]
The robot “remembers” where each motor was when the block was started.

KEY POINT: If the robot is moved before the touch button is pushed, the first motor movement will reset to its “initial” position plus movement. This causes random behavior. Use reset to clear motor memory.
Try this simple experiment. Implement the program below (or load *Random - Reset.rbt*). Move the robot forwards / backwards or turn it left / right, and then push the touch sensor.

The robot will not move forward consistently as specified by the move block block on right.
In the heat of competition, the team can forget to put the arm in the correct location.

Slow pulling back motion should not affect the robot’s forward movement.

Braking locks robot’s position.

Because movement is based on time, the robot may think that its arm is further back than it really is. Reset corrects the arm’s rotation angle.
• Notice how there are two tracks. NXT executes both tracks in parallel.

• So A motor pulls back arm as B-C motors move the robot forward.

• KEY POINT: Two parallel paths can’t drive the same motor at the same time. Therefore, the second path MUST terminate before the motor can be used in another path!
HOW TO CREATE PARALLEL TRACK

- Parallel tracks can be inserted anywhere.
- Simplest approach is to put block above or below spot in main track where it is to be joined.
- Hold shift button, then click mouse on track just left of block and drag to main track.
TEMPLE EXPLAINED #7: MOVE

- Start of mission
- Modify this block to move robot out of base.
- Add next robot steps to right.
TURNING TEMPLATE INTO MY BLOCK (STEP 1)

- Put mouse in upper left-hand corner, hold mouse down and drag to bottom right, highlighting everything.
TURNING TEMPLATE INTO MY BLOCK (STEP 2)

- Select “Make A New My Block” from Edit menu.
TURNING TEMPLATE INTO MY BLOCK (STEP 3)

- Change Simple Text to short, meaningful mission name.
- Click Next.
TURNING TEMPLATE INTO MY BLOCK (STEP 4)

- Pick unique icon that is indicative of mission.
- Click Finish.
BUILDING PROGRAM WITH MY BLOCKS

• Click double-blue bar to access custom palette.

• Click on upper left icon to select from My Blocks.
CREATE MASTER PROGRAM

- KEY POINT: By the time of the regional tournament, a master program can be created by stringing together the created mission blocks.

- KEY POINT: Add an initial block at the front of the master program that “waggles” the arm.
  - Train team to look for waggling when starting master program at robot performance table. This ensures that they didn’t start the wrong program.
CONSISTENT STARTS

• KEY POINT: If the start is not consistent, then the mission won’t run consistently.

• Two fundamental requirements for consistent starts:
  – Lining up robot exactly
  – Jerk-less initiation
LINING UP START

• KEY POINT: Slight differences in launch angle can make large differences where the robot ends up!

• In last year’s Food Factor, a 3° difference in going to the sink would shift the robot by nearly 3 inches.
REAR BUMPER

- KEY POINT: The easiest, fastest way to line up the robot is to push it back against the west or south wall.
- Of course, these walls need to be straight and meet at a right angle at base. Generally this is sufficiently true.
- Solution: Put long flat surface at back of robot.
LINING UP START (continued)

- KEY POINT: For each mission, pick a reference spot in base to align the left or right wheel. Fortunately, logos in base provide lots of reference spots. Examples:
  - Left wheel against the base top
  - Right wheel just above LEGO logo
  - Left wheel against bottom diamond
LINING UP START (continued)

- **KEY POINT:** The robot must be pulled back snugly against the wall prior to starting. Each team member must do this **every** time in the **same** way in practice and in competition.

- **How?** By grabbing the left and right motors – not the wheels – with the left and right hands.
JERK-LESS INITIATION

• KEY POINT: Press the touch sensor to start mission in such a way that alignment and motion aren’t affected. Each team member must do this every time in the same way in practice and in competition.

• How? By placing the forefinger and middle finger behind the touch sensor and the thumb on the touch sensor.
CREATING A MISSION

• Typically, missions require leaving base, going to another location and manipulating an object (e.g., pick up, drop off, trip lever).

• A good approach for creating a mission is to split into two efforts:
  – Navigate to location
  – At location, determine how to manipulate object
NAVIGATING TO A LOCATION

• Navigating is a series of turns and going straight defined distances.

• First, we need to characterize how the robot moves:
  – Number of motor degrees per inch
  – Number of two-motor degrees per rotational angular degree
  – Number of one-motor degrees per rotational angular degree
DETERMINE MOTOR DEGREES PER INCH

- Create a simple program that goes forward 1080 degrees (3 rotations).
- Measure distance.
- Degrees per inch = $1080° / \text{distance}$.
- For 81.6 x 15 motorcycle wheel = $1080° / 29.75" = 36.3$ degrees / inch.
DETERMINE 2-MOTOR DEGREES PER ROTATIONAL DEGREE

• Create a program that spins around itself exactly twice (720°). Initially, set BC motor to hard right-hand turn for 720 degrees. Adjust number of degrees until robot does exactly two rotations around itself.

• Note: The number of degrees will vary by size of wheels and spacing between motors.

• Motor degrees per rotational degree = motor degrees / 720°.

• For Sea Monster robot with 81.6 x 15 motorcycle wheel = 760° / 720° = 1.06° per degree.

• Example: To make a 30° left turn, set BC motor to 30° x 1.06 = 32 degrees.
DETERMINE 1-MOTOR DEGREES PER ROTATIONAL DEGREE

• Create a program that moves single motor so that robot spins around itself for 720°.

• Motor degrees per rotational degree = motor degrees / 720°.

• For Sea Monster robot with 81.6 x 15 motorcycle wheel = 1850° / 720° = 2.60° per degree.

• Example: To make a 30° right turn, set C motor to 30° x 2.60 = 77 degrees.
1-MOTOR VERSUS 2-MOTOR TURNS

• Did you notice that 1-motor turns finished in the same spot more consistently than 2-motor turns?
• Why? NXT does a great job in commanding individual motors to turn precise rotation angles. However, it doesn’t coordinate rotations of two motors simultaneously as precisely.
• KEY POINT: Use 1-motor turns for more precise control.
DEVELOPING A PROGRAM

- From Food Factor 2011 challenge, push bar to raise slide and catch rat when it falls off end of slide (15 points each).
- Attack problem by working out (1) how to retrieve rat from ramp and (2) how to maneuver robot to ramp.
CATCHING RAT

• From trying many different solutions, Sea Monsters found that a box attached to the robot’s front will catch rat.

• Box subsequently modified to enhance reliability:
  – High side and back walls to ensure rat falls into box.
  – Rail to extend front, otherwise light sensors on bottom would bump bottom of ramp and stop bar movement.
CATCHING RAT (continued)

- Sea Monsters found that pushing bar too strongly causes rat to fly off ramp and pushing bar too softly keeps rat from sliding down ramp.
- Repeated trials found that moving robot at 55% power into ramp bar caused rat to slide down ramp consistently.
- Trials also found that rat could take up to 2 seconds to slide down ramp to reach box.
CATCHING RAT (continued)

- **KEY POINT:** When moving the robot to collide with an object, use a **timed** movement.
- **Why?** If rotation movement is used instead and robot is physically stopped before it can complete wheel rotations, NXT will hang up trying to finish movement. Robot would then have to be rescued.
- **KEY POINT:** After colliding with an object, reset the motors.
- **Why?** If wheels slip after colliding with the object, NXT will adjust wheel rotations to compensate at the next movement, resulting in unpredictable behavior.
MOVING TO SLIDE

• A simple approach is to put small pieces of tape on the mat where you want the robot to go.
• Measure distance for each step and convert to degrees.
• For 66° turn, let’s assume that angle is estimated at 70°.
• Use conversion factors to determine each motor setting:
  – Straight 23.3” x 36.3°/inch = 846° (BC motor)
  – Left turn 70° x 2.6°/degree = 182° (B motor)
  – Straight 19” x 36.3°/inch = 691° (BC motor)

• Do first motor step. Verify lands on first tape mark (yes).
• Do second motor step. Verify points to second tape mark (close enough).
• Do third motor step. Robot goes too far to west. Adjust second motor step until lands on tape (172°).
USING VIEW

• An alternative approach to conversion is using the View feature on the robot to measure rotation. Procedure:
  – Select View
  – Select Motor Degrees
  – Pick Port A, B or C to match desired motor
  – Rotate wheel(s) to desired location

• Given that View is so simple to use, why did we spend the time in the previous slides to work out the relationship of distance / turn to motor degrees? Because it is important for every team member to have an intuitive feel for how the robot moves based on its programming. That way adjustments / corrections can be made directly as opposed to making repeated guesses.
RETURNING TO BASE

- Goal: Get back to base as fast as possible while avoiding table objects.
- Aim last leg to enter base in middle, and go deep into base to where robot can be easily picked up.
- Steps should be done close to maximum power.
- Last step should be done at 100%, timed and coast. Why? Time and coast will give robot enough momentum to reach base if angle is off.
PUTTING IT ALL TOGETHER

- *GET RAT.rbt* has final mission.
At this point, we now have a mission that picks up rat and returns it to base at least 90% of the time.

Are we done? No, not until the program runs as fast as possible while maintaining consistency. Admittedly, speeding up can wait until later when the need for reduced time becomes clear. However, every second is needed in competition, so have the team do it now while the program is best understood.

How to speed up?
- Go step-by-step and increase speed until a step loses consistency.
- Consider changing route on table to minimize distance traveled.
- Consider moving arm in parallel while robot is moving.
NOT DONE YET: SPEEDING UP MISSION (continued)

• KEY POINT: Do mission speed up on the “golden” robot.
• The golden robot is the one that performs best (most consistent, goes straightest).
• When motor speed goes up, two unmatched wheels or motors will cause the robot to pull increasing left or right, or wobble back and forth. So a mission that is optimized on the non-golden robot may not run well at full speed on the golden robot.
DELIBERATELY RESCUING ROBOT

• KEY POINT: Sometimes it is better to rescue the robot than have it attempt to return to base.
• Because position errors accumulate with distance traveled, the robot may end up far from base and can’t return reliably. Or the time it takes to return is more costly than simply picking up the robot, accepting the touch penalty and proceeding to the next mission.
DOING MULTIPLE THINGS

- **KEY POINT:** Don’t do just one thing in a mission; do multiple things.
- As we’ve noted before, time is the enemy. We therefore want to (1) minimize the number of times going to base and (2) increase the amount of things done each time out of base.
- So once a mission is fully working, ask the team three questions:
  - Now that the robot can do X, can it also do Y, which is nearby?
  - Can another mission be done with the same attachment?
  - Can the attachment be modified to solve multiple missions?
DOING MULTIPLE THINGS (continued)

• For example, in Food Factor 2011 there were four bacteria dispensers. After solving how to unload the green bacteria dispenser, the Sea Monsters worked out how to knock out the black pollution ball on the way back to base.

• Next, we developed programs to use the same attachment to unload the pink and red bacteria dispensers. Each required starting from base, but it took only 2 seconds to grab the robot, place against wall and press trigger button.

• Finally, the robot with the same attachment could be lined up again in 2 seconds and sent out to knock the yellow pollution ball and release the corn from the harvester.
DOING MULTIPLE THINGS (continued)

• In 2011 Food Factor, the Sea Monster robot did as its final mission flipping the thermometer (20 points), setting timer (14 points) and touching east wall (9 points) with the same arm attachment.
ALIGNING ROBOT DURING MISSION

• Most missions use “dead reckoning” to navigate around the mat. The robot has no feedback of its true location and orientation with each movement forward or turn. The position and orientation errors accumulate with each movement.

• There are a couple of ways for the robot to “find” itself.
  – Have the robot collide with a wall or object with a slow timed movement. Remember to reset the motors. Colliding with a wall does two important things:
    • Determines robot’s locations in one direction (north / south or east / west).
    • Squares the robot’s orientation.
  – Use sensors for navigating. The light sensors are particularly good in resolving both location and direction. Please refer to the Using Light Sensors training course.
BEING AWARE

• KEY POINT: Every time a mission is run in practice or competition, every team member present should pay close attention to every robot action.

• Should something unexpected happen, hopefully the clues are well enough remembered to aid in determining root cause.

• Second, watching carefully will often trigger ideas for improving missions.
  – For example, in the 2007 Power Puzzle challenge, the Sea Monsters struggled to move a truck back to base. On one occasion, the robot accidentally collided with the truck, causing it to point to base. Capitalizing on this, Sea Monsters added steps to “whack” the truck, reliably bringing it to base.
BEING AWARE (continued)

• KEY POINT: Don’t pick up the robot until every clue can be extracted.

• It is too easy for team members during practice to pick up the robot and run back to change the program.

• Don’t! Make sure that everything is understood before touching the robot. Should the robot have gone too far, not far enough or didn’t turn to desired orientation, or should the attachment not go high or low enough, picking up the robot makes this information vanish.

• Instead, use View or a ruler to determine how next to modify the robot.
TROUBLESHOOTING

• An easy trick is to put a button push or short timer between each step.

• Check that each step worked correctly and go to next step. Remove the button / timer after those steps that perform flawlessly.
TROUBLESHOOTING (continued)

- If the program still doesn’t work, consider trying other approaches.
- To keep from losing your work, don’t modify the baseline. Instead, make a copy and modify the copy.
- Don’t use “Save As” to save a copy of original. Why? If modifying a block, any program containing the original block will have it replaced with the modified block.
- Instead, in Windows Explorer make a copy and rename it. Mindstorms will treat the copy as a program and not as a block. If a variant becomes the desired program, then convert the program into a block.
BROKEN BLOCK

• If you see a zigzag through block as shown below, it means that file associated with the block can’t be found by Mindstorms.

• Solution: Locate block file from backup and put in Default -> Blocks -> My Blocks. Unfortunately, after restoring, the icon associated with the block will have been lost. Best to create a new block from the recovered program.
ATTACHMENTS & STRATEGIC OBJECTS

• First off, what is meant by attachments and strategic objects?
  – Attachments are objects attached to the robot to manipulate mission models. They are typically attached to the arm motor or to the robot body.
  – Strategic objects are standalone objects that are not attached to the robot. They are either delivered with mission objects and left outside of base, or used to prepare the robot before leaving the base such as an alignment tool.
ATTACHMENTS & STRATEGIC OBJECTS (continued)

• Key considerations:
  – Physically robust
  – Easy to attach to the robot and add mission models
  – Ideally, can be used in multiple missions
  – No pegs that can come off
  – Robot arm motor and front attachment points are firmly attached
ATTACHMENTS & STRATEGIC OBJECTS (continued)

- KEY POINT: All attachments and strategic objects must survive a drop test of 3’ to a hard floor with little or no damage.

- Why? In the heat of the competition, it is so easy to mishandle these objects, which could wreck the team’s performance.
  - In the 2007 VA/DC FLL Championship final round, the Sea Monsters dropped and shattered a strategic object that delivered 145 points worth of mission models. Thankfully, the Sea Monsters had earned the Division I high score in the previous round to win robot performance.
STRENGTHENING ROBOT FOR ATTACHMENTS

- KEY POINT: The robot must be designed so that the arm motor won’t pull apart. In the heat of competition, it is very easy for team to manhandle robot. Recommend using Technic cross block / fork 2x2 (Lego part no. W991405).

- KEY POINT: The robot must be designed so that front attachments can be put on and taken off without (1) damaging robot or (2) leaving pegs behind, which will cause confusion when adding the next attachment. Recommend using Technic beams with snaps (Lego part no. W991404). In fact, you can never own enough of these wonderful pieces.
STRENGTHENING ROBOT FOR ARM MOTOR

- The pictures below show how the Sea Monster robot attaches the arm motor to the NXT brick with the cross block / fork pieces. The arm motor won’t come off.
STRENGTHENING ROBOT FOR FRONT ATTACHMENTS

- Use Technic beam with snaps (#1 and #2) to connect with front attachments. This avoids issues with pegs.
- Strengthen with cross beams (#3 - #5).
CREATING ARM ATTACHMENTS

- **KEY POINT:** Use 3M friction connector pegs to attach to the arm motor. Wider ring in pegs should be sandwiched between two beams to prevent connector pegs from getting stuck in the arm motor when the attachment is removed.

- **KEY POINT:** Add another 3M peg to stop motion when the arm attachment is pulled back.
PIECES TO USE IN BUILDING ATTACHMENTS

- Use Technics bricks instead of standard Lego bricks. The Technics bricks snap together tighter and are therefore less likely to break apart.

- Use smooth top pieces to line bottoms of attachments to allow mission objects to slide out.
PIECES TO USE IN BUILDING ATTACHMENTS (cont’d)

- Put round pieces on bottom of front attachments that are pushed along mat. The round pieces minimize drag.
CREATING STRATEGIC OBJECTS

• KEY POINT: Use strategic objects to simplify delivery of mission objects.

• Pay close attention to the challenge. If it says that the mission object must touch the mat, be sure that the object has an open bottom.

• KEY POINT: Be sure to design the strategic object so that it can be easily lifted away at the end of the round, but leaves the scoring objects in place.

• In 2011 Food Factor challenge, the Sea Monsters created a simple box for returning the little fish to its original location. The front of the box had a raised wall, making it simple for the robot to push into place.
FOLLOWING WALL

• KEY POINT: Use side-mounted wheels to move along west or south wall. Advantages:
  – Robot can move fast and accurately along wall.
  – By adjusting the length of the side wheel attachments, the robot can be spaced precisely from wall, simplifying mission model manipulation.

• Notice how the attachment has box to prevent pieces from falling off table.
FOLLOWING WALL (continued)

• To follow wall, make slightest turn to right (south wall) or left (west wall).

• If no side wheel in back, then robot will have difficulty going backwards to return to base because robot is not aligned parallel to wall.

• Solution: Stop a few inches from mission object. Turn away from wall and move forward. Rotate robot so that it is parallel to wall before returning.
SECOND CHANCES

- **KEY POINT:** Look for opportunities to make multiple tries.
- In the 2011 Food Factor, the Sea Monsters found two such opportunities:
  - Delivering viruses to the sink. Putting nine or more viruses scored 13 points, while putting just one virus scored 7 points. With 16 viruses available, the Sea Monsters’ solution was to create an attachment with three delivery ports. The center one held up to 12 viruses, while the side ones held two. So if the robot came directly over the sink, it would get 13 points. If it were off by a couple of inches, it would still get 7 points.
SECOND CHANCES (continued)

– Moving timer’s hand to red zone. The Sea Monster robot gave itself two chances by positioning its arm to be just right of the timer’s center and pulling back to move the timer hand. The Sea Monster robot would then pivot left, move forward and pull back again. So if the original attempt was too far to the right, the second attempt would get it.

– Please notice below that the attachment had two contacts spaced 1M apart. So should the attachment go exactly over the center of the timer, the timer’s hand would go in between the two contacts.
NOT GOING STRAIGHT AT A MISSION OBJECT

• KEY POINT: Don’t go straight at an object if there is a risk of collision.

• Sometimes heading straight at a mission object can be risky where a slight error in heading will cause a collision. A good example is the 2011 Food Factor bacteria dispensers. The goal is to get a collecting attachment fully underneath the dispenser without colliding with the front of the dispenser.

• Solution: Steer the attachment safely to the right of the dispenser, then pivot into the dispenser using timed movement to trip the release.
ERRATIC BEHAVIOR

• KEY POINT: The single most likely cause of erratic performance is low battery voltage.
  – During practices, charge whenever convenient. Check battery voltage with monitor. Full battery should be around 8.4V.
  – During the competition, charge robot after every round.
• KEY POINT: Rechargeable batteries have limited lives.
  – Swap rechargeable batteries every 2 - 3 years.
  – If the rechargeable battery is suspected, replace the rechargeable battery with 6 AA batteries to confirm suspicion.
CONFIGURATION CONTROL

• KEY POINT: Always save the program every time before downloading! Otherwise, program changes can be lost.
  – Also, remember to open and save the master program that uses the block.

• KEY POINT: Back up the default folder before every practice!
  – Should a program revision make the program perform worse, can retrieve an earlier version.
  – Simplest is to go to Profile folder, copy Default subfolder and rename copy to “Default – Today’s Date”.
  – Periodically copy Profile to a thumb drive in case something happens to computer.
GOOD MINDSTORMS NXT REFERENCES

- Winning Design!: LEGO MINDSTORMS NXT Design Patterns for Fun and Competition by James J. Trobaugh.
- The LEGO MINDSTORMS NXT 2.0 Discovery Book by Laurens Valk.
WEB RESOURCES

- TechBrick Robotics (www.techbricks.com) is a great resource for early scoring sheets.
- The Lego Education Store (http://www.legoeducation.us/) should be your first stop for NXT and Technics accessories.
- The Lego Digital Designer software can be found at http://ldd.lego.com/.
KEY POINT SUMMARY

• The single most important thing for the robot to do is to go straight (page 4).
• Put a counterweight over the opposite wheel to balance arm motor (page 6).
• A mission isn’t finished until it works at least 9 out of 10 times in a row (page 7).
• At best, performance on competition table will match practice table, but often will be worst (page 8).
• Use the Sea Monster trick of turning each mission into a “block” (equivalent to a subroutine). Each block is triggered by pushing the touch sensor. (Page 12)
KEY POINT SUMMARY (continued)

• If the robot is moved before touch button is pushed, the first motor movement will reset to its “initial” position plus movement. This causes random behavior. Use reset to clear motor memory. (Page 17)

• Two parallel paths can’t drive the same motor at the same time. Therefore, the second path must terminate before the motor can be used in another path. (Page 20)

• By the time of the regional tournament, a master program can be created by stringing together the created blocks (page 28).

• Add an initial block at the front of the master program that “waggles” the arm (page 28).

• If the start is not consistent, then the mission won’t run consistently (page 29).
KEY POINT SUMMARY (continued)

- Slight differences in launch angle can make large differences where the robot ends up (page 30).
- The easiest, fastest way to line up the robot is to push it back against the west or south wall (page 31).
- For each mission, pick a reference spot in base to align the left or right wheel (page 32).
- The robot must be pulled back snugly against the wall prior to starting. Each team member must do this every time in the same way in practice and in competition. (Page 33)
- Press the touch sensor to start mission in such a way that alignment and motion aren’t affected. Each team member must do this every time in the same way in practice and in competition. (Page 34).
KEY POINT SUMMARY (continued)

- Use 1-motor turns for more precise control (page 40).
- When moving the robot to collide with an object, use a timed movement (page 44).
- After colliding with an object, reset the motors (page 44).
- Do mission speed up on the “golden” robot (page 51).
- Sometimes it is better to rescue the robot than have it attempt to return to base (page 52).
- Don’t do just one thing in a mission; do multiple things (page 53).
- Every time a mission is run in practice or competition, every team member present should pay close attention to every robot action (page 57).
KEY POINT SUMMARY (continued)

• Don’t pick up the robot until every clue can be extracted (page 58).

• All attachments and strategic objects must survive a drop test of 3’ to a hard floor with little or no damage (page 64).

• The robot must be designed so that the arm motor won’t pull apart (page 65).

• The robot must be designed so that front attachments can be put on and taken off without (1) damaging robot or (2) leaving pegs behind, which will cause confusion when adding the next attachment (page 65).

• Use strategic objects to simplify delivery of mission objects (page 71).
KEY POINT SUMMARY (continued)

- Be sure to design the strategic object so that it can be easily lifted away at the end of the round, but leaves the scoring objects in place (page 71).
- Use side-mounted wheels to move along west or south wall (page 72).
- Look for opportunities to make multiple tries (page 74).
- Don’t go straight at an object if there is a risk of collision (page 76).
- The single most likely cause of erratic performance is low battery voltage (page 77).
- Rechargeable batteries have limited lives (page 77).
- Always save the program every time before downloading. Otherwise, program changes can be lost. (Page 78)
• Back up the default folder before every practice (page 78).