DD2528 Assignment 1

Anne Haaker, Janne Schyffert, Jonatan Stagge, Phoebe Schwartz, Sofia Edvardsson

Smart Warehouse System Description

Task 1

1. A robot's arm is damaged.
2. Obstacles block the robot's path.
3. Robot collides with another robot
4. Robot collides with a shelf/wall
5. Robot collides with human operator
6. Robot drops box during loading or unloading.
7. A robot places the box incorrectly.
8. A robot has run out of battery.

Task 2

Name	**Pick up Box from Conveyor Belt**
Short description	A robot picks up a box from the conveyor belt.
Precondition	- There is a box on the conveyor belt. - An idle robot has been tasked to pick up the box. - The robot has moved to the conveyor belt.
Postcondition	The box is collected and on the robot's back.
Error situation	The robot fails to pick up the box.
System state in the event of an error	The box is not picked up, will require exception handling by a human.
Actors	Robot

Trigger	The WMS signals the robot to pick up the box.
Standard process	1) The conveyor belt stops moving. 2) The robot confirms the RFID of the box. 3) The robot grabs the box and places it onto its back.
Alternative process	1') The conveyor does not stop correctly, and the robot fails to grab the box. 2') RFID verification fails, robot does not grab box. 3') The robot fails to grab the box.

Name	**Transport Box to Shelf**
Short description	The robot delivers a box to a shelf.
Precondition	- The robot is carrying a box. - The box has been assigned a place on a shelf. - The WMS has planned a route to the shelf for the robot and sent it to the robot. - The robot has enough battery to reach the shelf.
Postcondition	The robot is located at the designated shelf.
Error situation	The box is not transported to the shelf.
System state in the event of an error	The box remains on the robot.
Actors	Robot
Trigger	WMS sends a planned route to the robot.
Standard process	1) The robot moves to the designated shelf, continuously looking for obstacles.
Alternative process	1') The robot fails to move to the shelf, reports an error to WMS

Name	**Check battery status of Robot**
Short description	The WMS queries a robot for its battery status.e

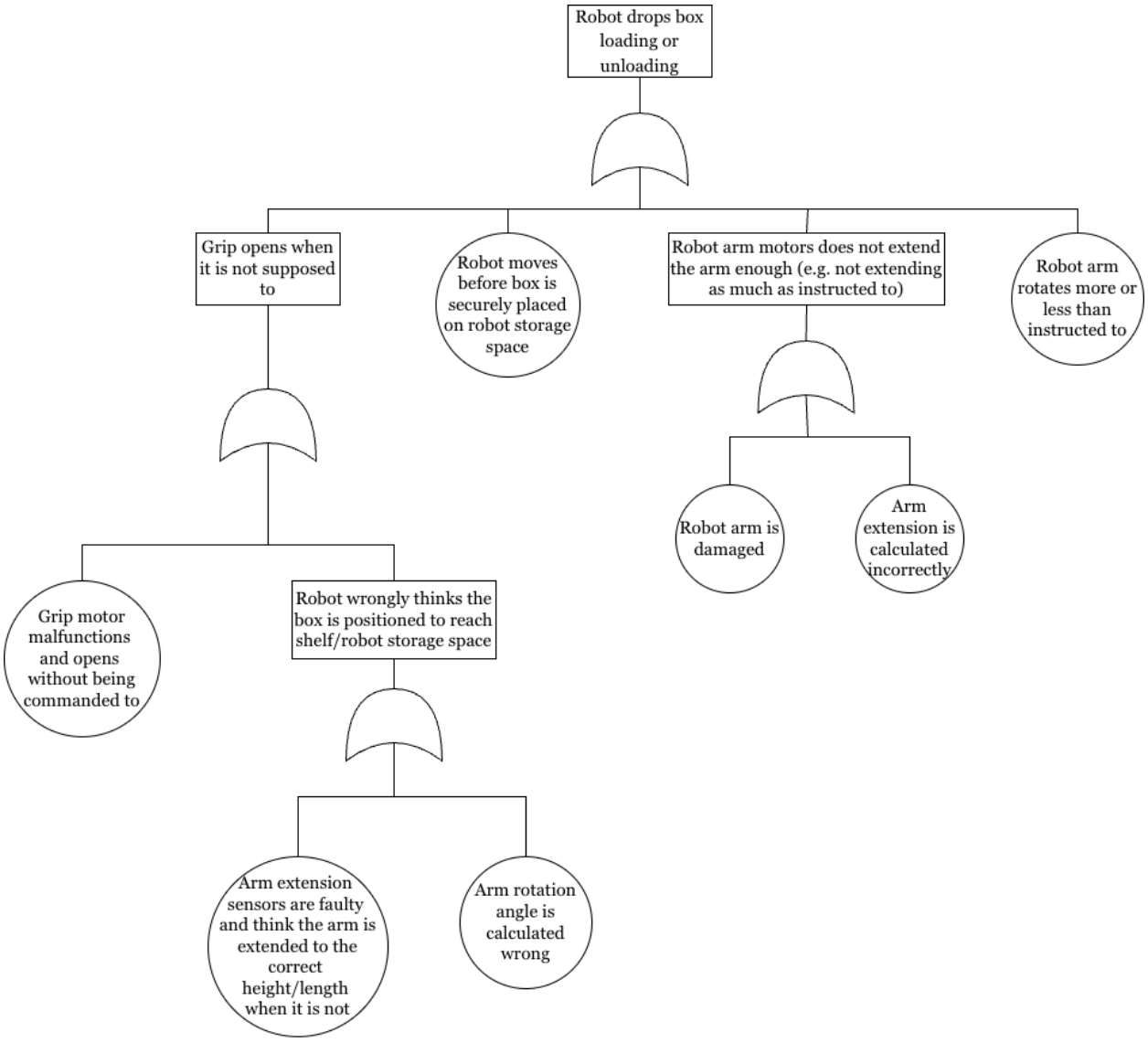
Precondition	- There is a robot selected for the query. - The robot has queried its sensor for its current battery status.
Postcondition	The WMS is aware of the robots current battery status.
Error situation	The WMS is unaware of the robots battery status.
System state in the event of an error	The robot cannot be assigned tasks.
Actors	WMS
Trigger	The WMS wants to designate the robot for a task.
Standard process	1) WMS queries the robot for its battery status. 2) The robot communicates its battery status to the WMS. 3) The WMS notes that the battery status is sufficient, and can proceed giving the robot a task.
Alternative process	2') The robot fails to communicate its battery status. WMS flags exception. 3') The WMS notes that the battery status is low, and sends the robot to a charging station.

Name	**Place Box on Shelf**
Short description	A robot drops off a box at its designated shelf.
Precondition	- The robot has moved to the shelf. - The robot has a box in its storage container.
Postcondition	The box is on the shelf.
Error situation	The robot fails to place the box on the shelf.
System state in the event of an error	The box remains in the robot's storage container.
Actors	Robot
Trigger	The robot has finished transporting the box to the shelf.
Standard process	1) The robot grabs the box from its storage

	compartment. 2) The arm is raised/lowered to the correct height. 3) The box is placed on the shelf.
Alternative process	1') The robot fails to grab the box. 2') Arm movement fails. 3') The robot fails to release the box.

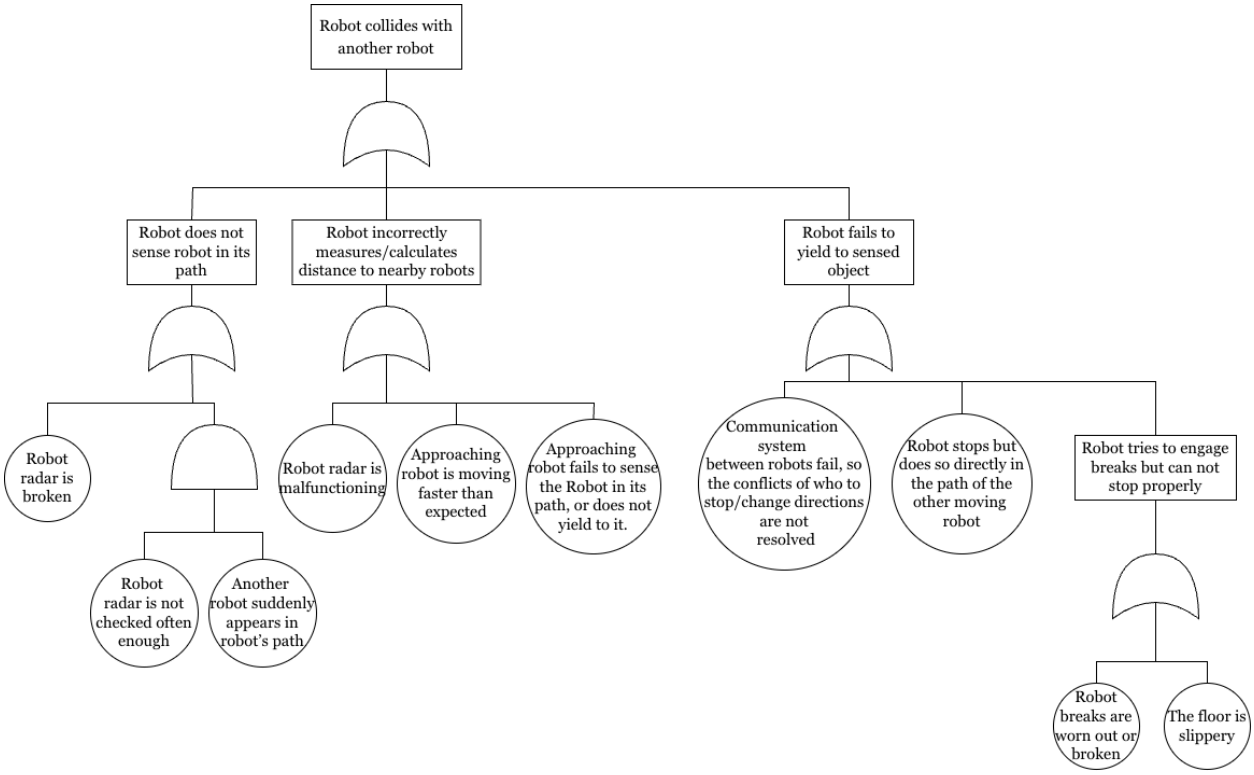
Name	**Recharge Battery**
Short description	The charging station recharges the battery of a robot.
Precondition	- A robot is in the charging station - The robot in the charging station is not fully charged.
Postcondition	The robot in the charging station is fully charged.
Error situation	The charging station fails to recharge the robot.
System state in the event of an error	Robot is blocking a charging station, needs manual handling
Actors	Charging station
Trigger	A robot enters the charging station
Standard process	1) The charging station charges the robot
Alternative process	1') The charging station fails to charge the robot

Task 3



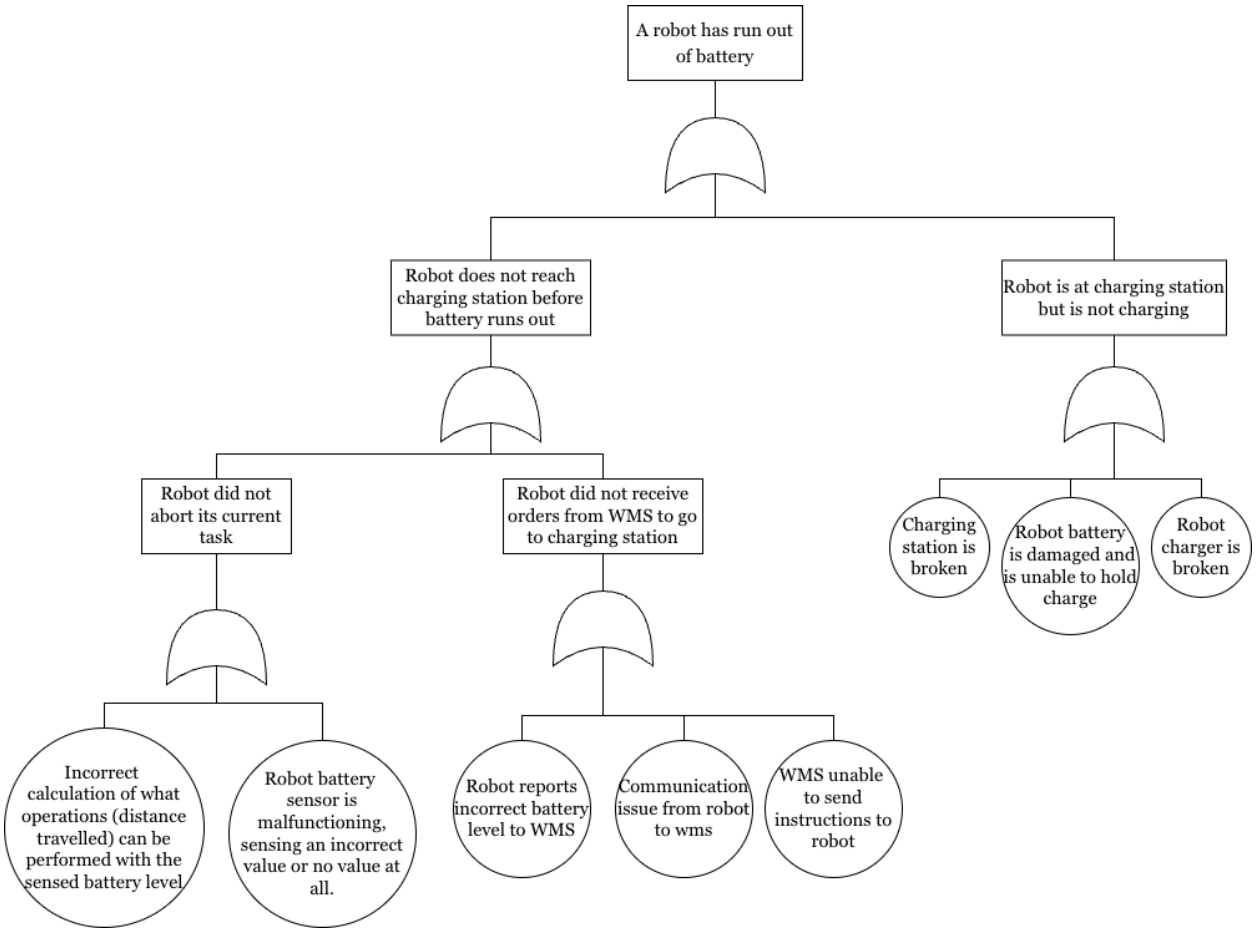
Robot drops box loading or unloading

- Grip opens when it is not supposed to
    - Grip malfunctions and opens without being commanded to
    - Robot wrongly thinks the box is positioned to reach shelf/robot storage space
        - Arm extension sensors are faulty and think the arm is extended to the correct height/length when it is not
        - Arm rotation angle is calculated wrong
- Robot moves before box is securely placed on robot storage space
- Robot arm motors does not extend the arm enough (e.g. not extending as much as instructed to)
    - Robot arm is damaged
    - Arm extension is calculated incorrectly
- Robot arm rotates more or less than instructed to



Robot collides with another robot

- Robot does not sense robot in its path
    - Robot radar is broken
    - Robot radar is not checked often enough AND another robot suddenly appears in robot's path
    - AND another robot is approaching and fails to sense the Robot in its path, or does not yield to it.
- Robot incorrectly measures/calculates distance to nearby robots
    - Robot radar is malfunctioning
    - Approaching robot is moving faster than expected
- Robot fails to yield to sensed robot
    - Communication system between robots fail, so the conflict of which robot should stop/change path is not resolved
    - Robot stops but does so directly in the path of the other moving robot
    - Robot tries to engage breaks but can not stop properly
        - Robot breaks are worn out or broken
        - The floor is slippery



A robot has run out of battery

- Robot does not reach charging station before battery runs out
    - Robot did not abort its current task
        - Incorrect calculation of what operations (distance travelled) can be performed with the sensed battery level
            - Robot battery sensor is malfunctioning, sensing an incorrect value or no value at all.
    - Robot did not receive orders from WMS to go to charging station
        - Robot reports incorrect battery level to WMS
            - Robot battery sensor is malfunctioning
        - Robot unable to send battery level status to WMS
            - Communication failure
        - WMS unable to send instructions to robot
            - Communication failure
- Robot is at charging station but is not charging
    - Charging station is broken
    - Robot charger is broken
    - Robot battery is damaged and is unable to hold charge

Task 4.1

SSR1: When a human enters the warehouse, they should stop in a specific and marked position where the robots cannot go to avoid being run over. The robots should pause their current task until the human has left the warehouse.
SSR2: Robot arm should be retracted before robot starts moving, to avoid damaging the arm.
SSR3: Robot radar should be reliable and should be checked often, to avoid collision with other objects such as other robots or walls.
SSR4: The communication system between robots and WMS should ensure timely, reliable and error-free transmission and reception of data (commands, robot coordinates, battery status, status of task).
SSR5: Robot chargers should be checked often to ensure working condition.
SSR6: The robot's battery level should regularly be queried, and the predefined function to calculate the possible travel distance based on this battery level should be used to ensure enough battery charge for requested task.
SSR7: WMS system should have strong processing power and memory to keep track of and organize the robots, tasks, boxes, and errors.
SSR8: Readings from the robot's sensors are validated against predicted values whenever the corresponding functionality is used, e.g. the arm extension sensor value is compared against a predicted value based on movement over time to ensure accuracy and correctness of sensors.

SSR9: Robot mechanics, such as arm extension and grip, are checked often by humans to ensure that the mechanical parts of the robot are behaving as expected.
SSR10: The robot should abort its current task and move to the closest charging station before the battery level becomes critically close to the lowest battery charge needed to reach a charging station, as calculated based on the predefined function that calculates possible travel distance based on battery level.
SSR11: The robot should verify that the box's RFID tag is the same as the one provided by the WMS for this task, before placing the box on the shelf or conveyor belt. If the tags don't match, there should be error handling where the robot stops the task and reports to WMS.

Task 4.2

- Register incoming box: SSR7
- Assign storage location: SSR7
- Choose idle robot: SSR6, SSR7
- Check battery status of robot: SSR4
- Send robot to charging station: SSR4
- Plan route: SSR4, SSR7
- Monitor exceptions: SSR4, SSR7

- Resolve exceptions: SSR1

- Recharge battery: SSR5, SSR10

- Receive route: SSR4
- Detect obstacles: SSR3
- Move to collection point: SSR2, SSR3, SSR6
- Transport box to shelf: SSR2, SSR3, SSR6, SSR8, SSR9
- Deliver box to conveyor belt: SSR6, SSR8, SSR9
- Move to charging station: SSR2, SSR3, SSR6, SSR10
- Verify RFID: SS11
- Pick up box from conveyor belt: SSR6, SSR8, SSR9
- Place box on shelf: SSR6, SSR8, SSR9
- Retrieve box from shelf: SSR6, SSR8, SSR9
- Send status update: SSR4
- Report failure: SSR4

Task 5.1

Component	**Sensor Measuring Height**
Failure Mode	Sensor measures too high or too low
Possible Cause	Hardware failure
Local Effect	The sensor measures a too high or too low value, so the measurement does not correspond to real life.
System Effect	The robot arm does not accurately reach the intended target height, potentially causing box to be dropped. The arm can be damaged if extended too much or colliding with object such as a shelf.
Detection	Software detection: Predict the height of the arm based on height of arm before extension/retraction and the time the vertical extension motor was turned on. Compare the predicted height of the arm to the sensed value to check if it is within error tolerance.
Remedial Action	Stop task execution. Notify WMS of a hardware failure and send human to check robot. Human operator should ensure ensure the warehouse is in a safe state (i.e. malfunctioning robot is removed or fixed, and potentially dropped box is removed). Afterwards, WMS should send a new robot to complete the task.

Component	**Sensor Measuring Extension Length**
Failure Mode	Sensor measures too far or short
Possible Cause	Hardware failure
Local Effect	The sensor measures a too-far or too-short value, so the measurement does not correspond to real life.
System Effect	The robot arm does not accurately reach the intended target length, potentially causing box to be dropped or not picked up at all. The robot arm can be damaged if extended too much or colliding with another object, such as a shelf or another robot. Another object in the warehouse may be damaged if the extended arm collides with another robot or knocks a box off a shelf.
Detection	Software detection: Predict the length of the arm based on previous length and the time the horizontal extension motor was turned on and compare this to the sensed value to check if it is within error tolerance.
Remedial Action	Stop task execution. Notify WMS of a hardware failure and send human to check robot. After human operator ensures the warehouse is in a safe state (i.e. malfunctioning robot is removed or fixed, and potentially dropped box is removed), WMS should send a new robot to complete the task.

Component	**Grip Motor**
Failure Mode	Grip is stuck on open or closed
Possible Cause	Mechanical failure, hardware failure or software failure
Local Effect	The grip does not correctly open or close when commanded to do so, staying stuck in the previous state.
System Effect	The robot fails to grab or drop a box when supposed to, or inadvertently drops a box.

	The robot is unable to complete its assigned task of picking up or putting down a box.
Detection	The grip sensor does not change value when the grip is commanded to close or open, indicating that the grip did not successfully open or close.
Remedial Action	Stop task execution. Notify WMS of a failure and send send human to check robot. Human operator should remove or fix faulty robot, and ensure the warehouse is in a safe state (such as removing dropped boxes). WMS should send a new robot to complete the task.

Task 5.2

Name	Fetch a box from height h
Short description	Robot picks up a box from height h and places it in its storage space
Precondition: prerequisite for successful execution	There is a box at height h. The robot storage has enough space for the new box. The robot is in front of the location of the box and can reach it. The robot has enough battery to complete the required arm actions of extending/retracting arm laterally and vertically, rotating the arm and closing/opening the grip.
Postcondition: system state after successful execution	Box is in the robot's storage space.
Error situations: errors relevant to the problem domain	The arm cannot extend to needed length. The grip malfunctions (does not open/close when commanded). The arm cannot reach height h. The robot runs out of battery, unable to complete the task. There is no box at the expected location.
System state on the occurrence of an error	The box is not fetched from height h.

Actors that communicate with the use case	WMS, robot
Trigger: events which initiate/start the use case	There is a box at height h. WMS sends a task to an idle robot to fetch the box at height h. The robot moves through the warehouse to the location of the box.
Standard process: individual steps to be taken	1. Robot's arm is extended to height h 2. Check for possible errors: a. Calculate expected value of arm height b. Compare sensor value of height measurement to expected height. If within tolerance, continue. Else, go to the alternative process. 3. Robot's arm is extended to the needed length. 4. Check for possible errors: a. Calculate expected value of arm length b. Compare sensor value of length measurement to expected length. If within tolerance, continue, else go to the alternative process. 5. Grip motor is activated. 6. Check for possible errors: a. Sensor should report grip as open before the motor is activated, then it should report the grip as closed. If the sensor status does not change, go to the alternative process. 7. Rotate arm to correct angle above storage space. 8. Adjust arm extension to the needed length to reach storage space 9. Grip motor is deactivated. Box is dropped. a. Sensor should report grip as closed before the motor is deactivated, then it should report the grip as open. If the sensor status does not change, go to the alternative process. 10. Box is on robot's storage space. a. Check if weight sensor value for the robot's storage space has increased, indicating

	successful placement of box on storage space. If no increase, go to alternative process.
Alternative processes: deviations from the standard process	1. (step 2, 4 or 6) Error detected. 3. Robot sends error message to WMS 4. WMS sends robot to time out and requests human to intervene. 5. WMS asks another robot to complete the task to fetch box from height h.

Task 5.3

Link to GitHub Repository with pseudocode implementation

fetchBox (h)

$p_v \leftarrow$ last known arm height
$t_v \leftarrow$ motor activation time calculated from $h - p_v$
activate vertical motor for t seconds
$s_v \leftarrow$ get vertical arm height from sensor
correct $\leftarrow$ verifyArmHeight(s_v, H)
if-not *correct*:
  return "Vertical Arm Extension Error!"

$p_l \leftarrow$ last known arm extension
$t_l \leftarrow$ motor activation time calculated from $h - p_l$
activate lateral motor for t seconds
$s_l \leftarrow$ get lateral arm extension from sensor
correct $\leftarrow$ **verifyArmExtension**(s_l, H)
if-not *correct*:
  return "Lateral Arm Extension Error!"

$g \leftarrow$ sensor grip state
if $g =$ closed
  openGrip()
closeGrip()
$g \leftarrow$ sensor grip state
if $g =$ open
  return "Grip Error!"

rotateArm()
adjustLateralExtension()
$p_l \leftarrow$ current arm extension from sensor

openGrip()
$g \leftarrow$ sensor grip state
if not $g =$ open
  return "Grip Error!"
$s \leftarrow$ Storage space sensor
if-not s = occupied
  return "Uh oh! There is nothing on my back"

return (height $= h$ extension $= p_l$ grip $=$ open backWeight $= s$)