Group 5 - Evan Laflamme (260684099) and Zhiyu Chen (260668145)

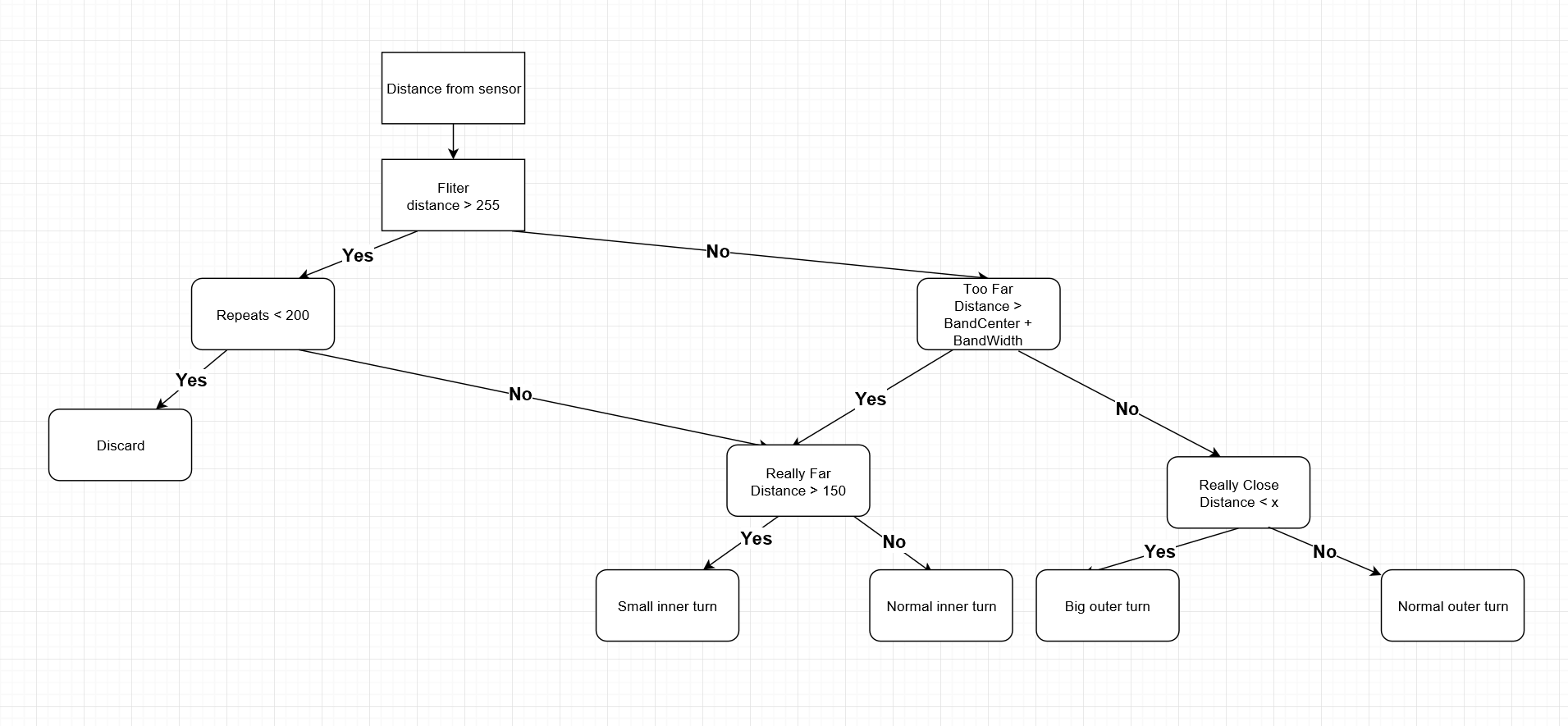
ECSE 211 – Lab 1 Report

* Design Evaluation:

Software

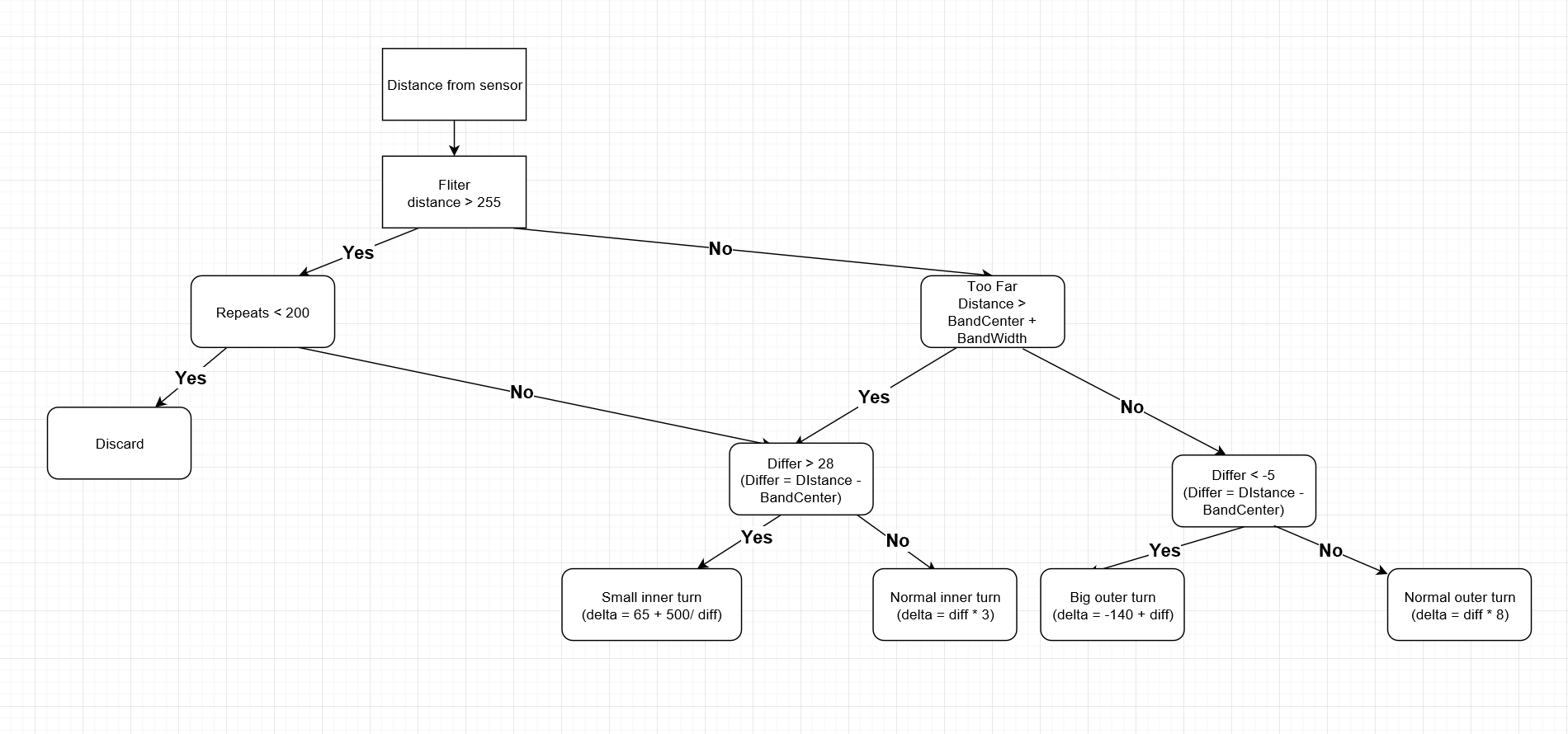
BangBang Controller

To implement a BangBang controller we follow the basic concept. Using +/- operations to adjust the relative position against the Wall. The flow chart below shows how we implement.



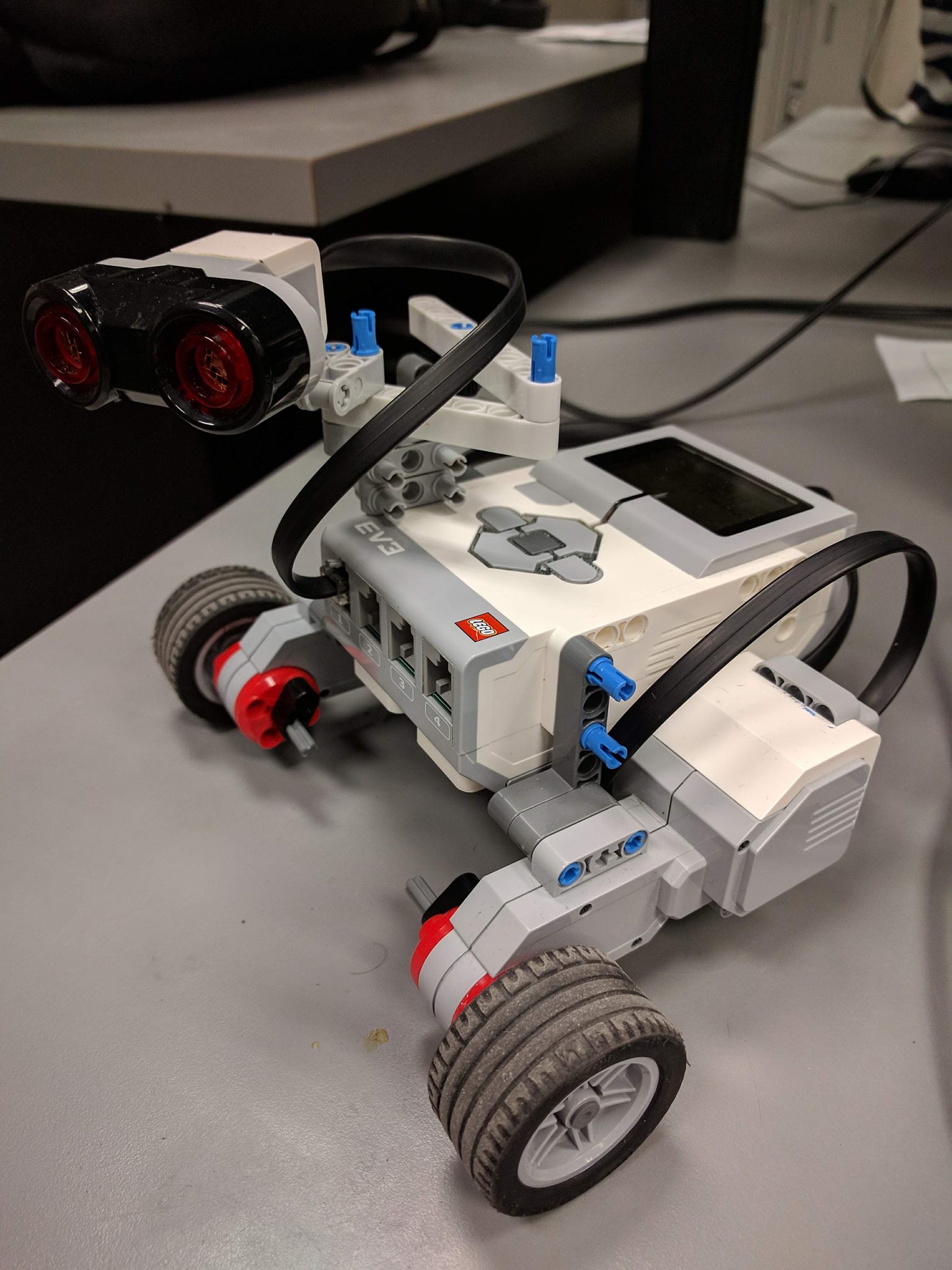
P controller

For P controller we change the wheel speed by giving it a multiplier and add a threshold to prevent the wheel reach maximum speed when the distance is really far. When applying the threshold we also give it a proportional increase to make it follow the concept of P controller.



Hardware

We use two motors to drive our car and put the sensor at about 55 degree so that it can detect the obstacles in front of it.



* Test Data:

|  |  |  |
| --- | --- | --- |
|  | Bang-Bang Controller | P-type Controller |
| Trial 1 | The robot’s distance from the wall is fairly constant. Robot oscillates rigidly in order to reposition itself. Convex corner turns are quite smooth. Concave corner turns require several adjustments. The robot successfully ignores gaps. | The robot remains at an appropriate distance from the wall with a few smooth adjustments. When the robot performs a turn at a convex corner, it rotates in place to adjust its orientation. Concave turns are very smooth. The robot ignores all gaps. |
| Trial 2 | The distance from the wall remains constant along straight edges. The robot adjusts quite a bit to remain a good distance from the wall especially after encountering a corner. The corner turns are smooth with few adjustments. The robot ignores all gaps. | The robot had to adjust slightly and smoothly in order to keep a good distance from the wall. The robot is required to rotate itself in place to avoid getting too close to the wall. Concave turns require a few slight adjustments to stay at a good distance from the wall. All gaps in the track were ignored by the robot. |
| Trial 3 | The robot remains at a constant distance from the wall with only a few quick adjustments. Convex turns are smooth and concave turns require slight adjustments to maintain good distance from wall. The robot ignored all gaps. | The robot makes smooth adjustments in order to reposition itself along the wall especially after a turn. The robot must rotate itself at convex corners to avoid the wall. The robot performs smooth concave turns. The gaps in the track are ignored. |

* Test Analysis:

The bang-bang controller is able to keep the robot at a fixed distance from the wall within the band width. It adjusts itself quite a bit to do so however, especially after the robot performs a turn. These adjustments are most prominent when performing a concave turn.

The design of both controllers ensures that whenever the robot is within the band width, it will simply move forward. This, however, means that the robot may oscillate from one side of the band to the other. For the bang-bang controller, the robot will adjust with a fixed speed in order to re-enter the band. This means that the robot may make quick, rough adjustments. For the p-type controller, the robot makes adjustments based on its distance from the wall. The design of the p-type controller ensures that when the robot is quite close to the band, it will make small, smooth adjustments to re-enter the band. This makes the robots adjustments and oscillations from one side of the band to the other much smoother compared to the bang-bang controller.

* Observations and Conclusions:

The ultrasonic sensor that comes with the Mindstorm kit has a few issues regarding reliability. It often detects incorrect readings. Since the robot is designed with the ultrasonic sensor at an angle, the ultrasonic waves it propagates get reflected off the smooth wall. This leads to random large spikes in the readings. These errors can be easily filtered out using a filter similar to that provided with the sample code. The filter will ignore large readings for a specified amount of time until it becomes clear that the large readings are valid.

False positives are produced by the sensor. For example, if the robot is moving towards a convex corner, the ultrasonic waves can reflect on the walls in order to make the robot think it is in fact closer to the wall than it actually is. The sensor also experiences several false negatives for example with large spiked values making the robot think that the wall is far away. As previously explained, these false positives can easily be filtered out.

* Further Improvements:

Physical design:

1. In considering the limiting data we now can get from the fixed sensor. I suggest putting a motor on the US sensor making it rotates constantly within a giving range to get more distance data from surroundings and avoid collision.

2. For now we always get wheel slipping during our experiment. To solve this I think we should decrease the length of wheel-base and add more weight to the overall structure to achieve a more flexible turning.

Software design:

1. For the P controller, I think we could use the property of exponential function to achieve a proportional change instead of using a constant multiplier and adding a threshold after.

Other types of controller:

1. Sliding mode controller

Widely adopted in nowadays UAV system. Can provide many advantages like precise tracking and robustness against disturbance and unpredicted inaccuracies.

1. PID controller

Being used in cruise control on vehicles. Can adjust the corrections according to the surroundings. Provide a smoother control than the P type. May have us solve the wheel slipping issue.