Have any one heard about industry 4.0 or Intelligent Manufacturing? In fact, right now, the manufacture industry is facing a new revolution. With the help of critical technology, we can make many products in incredible speed and high quality. To do so, the autonomous mobile robot or AMR plays a crucial role in this revolution. There are several reasons. The most essential thing is that AMR save the labor cost and increase the production rate dramatically. Further more, many applications of AMR have been found in other field such as medical services and military. There are some reports also indicate that the progress rate of AMR market was predicted to clime to 25% form now to 2029. We can expect that those robots will probably reshape our working environment in the years to come.

This will bring to a problem. In the future, many people need to work with AMRs. This will not just happen in advanced chip factories, but also in hotels, restaurants or even on a street. Therefore, those robots must have the ability to avoid humans or cars. But, unfortunately, most of cutting-edge AMRs can’t avoid all kinds of moving obstacles. This mean that the navigation system in robots still have space to be improved.

What my research focus on is trying to develop an obstacle avoiding algorithm that working on a specific robot type—general bicycle model or GBM. Just like bicycles, this robot has two wheels. But what make them different is those two wheels can spin and rotate independently. So, unlike normal bicycles which can only go forward and change their direction by their front wheel, the GBM can adjust their moving direction without changing its orientation. This crab-like action gives GBM a lot of maneuverability to move around. There is another question. How can I make this robot automatically avoid obstacles and reach a goal? What I want to do is making the robots develop a kind of ‘special awareness’ by consistently observing obstacles around them and counting the potential risk of collision. Thus, if the GBM want to find the safest path to go forward, this robot needs to keep adjusting its decisions and try to minimize the collision risk in every time period.

Actually, if we want to make a robot have the ability to evaluate the risk around it, we have to build a complex math equation to describe those risk. But intricate math formulas make the robot’s computer hard to optimize a best decision and consequently, it stuck in there occasionally. This is a dangerous situation that nobody wants it to happen. So, simplify the objective function and constraints is the issue I am facing now. Provided the problem is solved, we could get closer to a fully automatic AMR in the future.