4.

I choose the sheepdog bot. To design it, it should possess some abilities to deal with some issues. The main goal of sheepdog is leading sheep safely from the stable to pasture. To achieve this purpose, it should deal with safety concerns, control issue, and efficiency.

For the safety, firstly it should have ability to detect the obstacles and predators. The obstacles and predators are quite obvious. It could be solved by the neural networks. We need to collect the image of sheep, the image of the predators, such as wolf, and the image of the obstacles, such as huge rock or vines. Then, by using neural networks, we can find the patterns to distinguish sheep from the others and detect the predators, and obstacles.

Also, the sheepdog should have ability to find the best spot. The best spot means the place that have more foods for sheep, and which is safer than other places. Because the pasture is huge, it is better to assign the district. The size of each district should be determined by the number of sheep and the time they spend to eat all foods in that district. By using the ability to detect obstacles and predators, it could collect the data of them and assign the probability of the safety. Also, we got an information of remain foods in each district. The expected value of each district could be calculated by multiplying the probability of safety to remain foods. Based on that expected value, it can choose the next best district.

For the control, firstly it should know how to handle the sheep. Not all sheep are following well. So if any sheep not follow or move in wrong direction, it should handle them to follow. There are several ways to make them follow, such as blocking, biting, barking (threatening), and pushing. Blocking is the weakest way, biting (softly, no blood) is stronger, barking is much stronger, and pushing is the strongest way. The strongest way is better way to handle, however it possess possibility to panic them. We can collect the data after some trials, then get some probabilities, P(handled | blocking), P(handled | biting), P(handled | barking), P(handled | pushing),

P(panic | blocking), P(panic | biting), P(panic | barking) and P(panic | pushing).

The whole probability could be divided into two part, H(handled) and NH(not-handled). NH could be divided into two part, P(panic) and NP(not-panic). If we assigned the probability of blocking, biting, barking, and pushing, we can get the probability of H, NH, P, and NP. Our goal is reduce the P and increase the H. We can expect the value based on weighting the cost of H and P, and tried the several probabilities of each action. Then we can find the best way to handle the sheep.

The other required ability is to check the sheep's status to deal with them better. There are several states such as, panic, tired, sleepy, hungry, and angry. There could be more possible states, but just take those 5 as example. We will use knowledge base logic to check the states. We only can infer the status by using the observable (visible) knowledge. One of the usable observations are their motion, such as sitting, standing still, moving, running, and eating. Also the sound is observable, such as crying, breathing (dyspnea), and being silent. The other possible observation is any injury, and the tear. We can possibly infer the status by using these observations. For example, if the sheep is injured, crying, and running, it would be the panic status. It could be expressed as (Injured ^ crying ^ running) -> panic. Like this, we can build complex knowledge base to determine the sheep's status.

Because it is a robot, it has a issue of battery. Thus, it is better to reduce the problems, avoid the battery consuming situations, and be as efficient as possible. To reduce the moves, it is better to possess the ability to stand in the best position. If we assume this bot could calculate the distance between sheep, then it could stand in the middle of them. However, it costs a lot and almost impossible to get the information every time they move. Thus, using real sheepdog would be better than this bot. Instead, it could possibly detect the farthest sheep by using the detection I mentioned above (neural networks). Based on that image size, it can possibly get the information of its distance. If the sheep is out of certain range, then the bot would move and try to handle the sheep.

Also, to reduce the time for the battery, it is better to reduce the time to travel. Thus, it is better to find the shortest path from stable to pasture. Because the road is usually unpaved, it cannot use the gps. The possible way is that using informed search. Firstly, divide the district and decide the path to each district. Then calculate the time and the distance it took for traveling. Try the all possible route as possible. Based on this information, we could find the shortest way in point of time-spent and distance.

In summary, this bot should possess the ability to :

1. Detect the obstacles or predators 2. Find the safe spot 3. Control the sheep 4. Check the sheep's status 5. Find the best position to stand 6. Find the shortest path