Abstract

Quadcopters are likely to be used for transportation job in future. In the absenteeism of any hurdle or barrier, satellite located navigation, it would be an easy task to navigate from initial state to geostationary based leaving state (final state). Contrary to this scenario, in the presence of any hurdle or barrier in route of drone, flyer should come up with a flight scheme to repudiate any proposed hurdle or barrier. In case of unknown environment, there would be many unknown barriers that would be placed on unknown locations, then it would be a challenging task to come up with an appropriate flight plan. In addition to this scenario, in a low satellite communication surrounding, like in indoors scenario, under forestry or tree(urban) environment, so quadcopter navigation plan will be failed in this scenario. To tackle this problem, we can come up with the solution known to us AI (Artificial intelligence), the most emerging name in science and technology. AI can help robotics to perform under uncertain circumstances and overcome navigation problems. AI was in start, used to train robots, but in this era, it has been helping us to train ground vehicles, autonomous cars, and now begins to train quadcopter as well. Reinforcement Learning is one the best and emerging type of Artificial intelligence. Reinforcement Learning indoctrinate our model to perform desired result in a designed environment. The implementation of Reinforcement Learning on Quadcopter enhances the intelligence capability of it and makes it a fully autonomous robot. In this study first need to address the problem of hurdle avoidance of an autonomous UAV, which need to be trained to avoid all the hurdle/obstacles that comes in trajectory of it. Initially, we introduced a notion of to track our robot to follow the defined trajectory, to reduce Euclidean distance from the defined trajectory (likewise in line follower robotics), which would be allowing quadcopter to follow the defined trajectory, based on previous work. Euclidean distance strategy would be used to generate navigation strategy for UAV to maintain it along the trajectory. To account the scenario of hurdle avoidance of UAV (unmanned autonomous vehicle) in an unknown environment via reinforcement learning, we have tackled down our problem into two feasible parts. First would be hurdle avoidance, that would allow our robot to sense the environment (unknown to our robot). Once our robot sense hurdle, then our robot needed to take necessary action steps to avoid that hurdle via reinforcement learning. All hurdle avoidance and necessary action steps decision would be dependent on reinforcement

learning (Deep Q-learning). For hurdle avoidance, we need to train our model via object detection, training through deep learning, once hurdle has been detected our model will be taking necessary steps of action to alter or change its next state. Our main objective in this work is that to train our model to avoid obstacles coming up in its path and following our designated path. Now the question arises, from where our model gets environment information, from where our model will be sensing the environment. Physically quadcopter have two main sensors, one is gyro-sensor (motion detection sensor) and the other one is quadcopter's stereoscopic based front camera. Gyro would be lying under the category of external sensor and would have frequency ranges issue, so it would not be a feasible solution to opt gyro-based sensor in current scenario. One of the most important reason behind selecting the stereo vision front camera of our model, we can get all the deep and minor information lying under our environment. The objective is to train our model in an environment that include hurdles, we must design reward function for it, to get the optimize path. Frame of reference would be defined, to keep our model within the predefined limits, or that would not t allow our model to leave or enter the defined frame. Our quadcopter should be trained to circumvent the hurdle within predefined frame or area and should land to its destination. Our results show that in differentiation to earlier work, the advanced algorithm have much preferable than the existing one, not only in the context of our reward function, but also in much effective time. The second achievement is for checkpoints, that provide us the opportunity of storing a trained model for each time step, to achieve a better reward from the preceding one. This flexibility allows us not to train our model for same check points again and again in the long run. Our result shows the flexibility of checkpoint to get better test results.