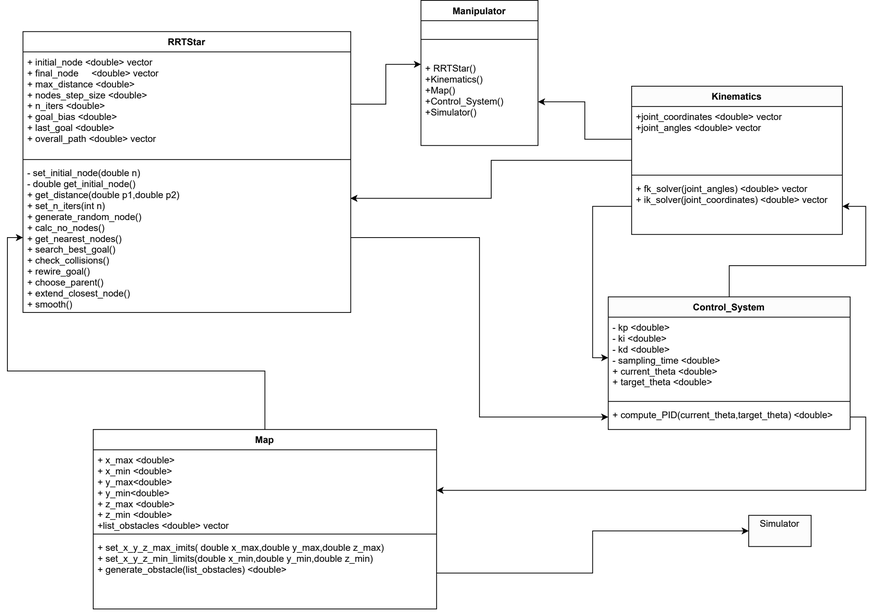
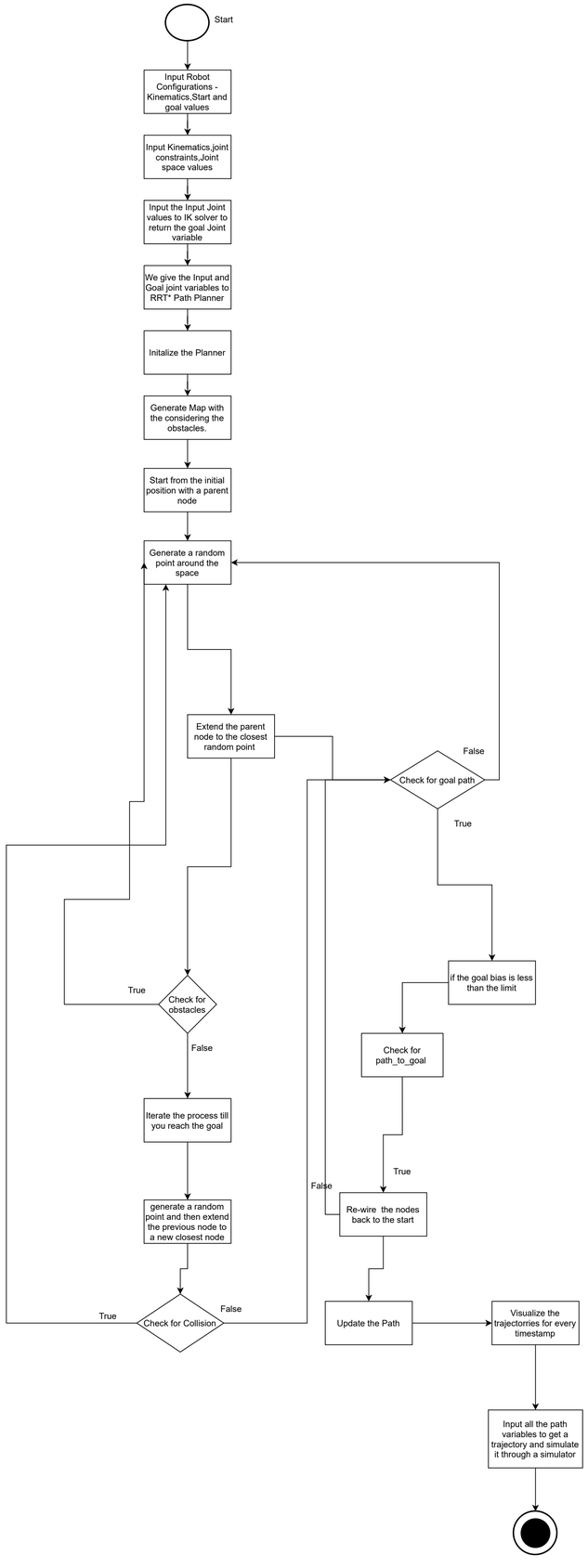

Manipulator
+ RRTStar()
+Kinematics()
+Map()
+Control_System()
+Simulator()
RRTStar
+ initial_node <double> vector
+ final_node <double> vector
+ max_distance <double>
+ nodes_step_size <double>
+ n_iters <double>
+ goal_bias <double>
+ last_goal <double>
+ overall_path <double> vector
- set_initial_node(double n)
- double get_initial_node()
+ get_distance(double p1,double p2)
+ set_n_iters(int n)
+ generate_random_node()
+ calc_no_nodes()
+ get_nearest_nodes()
+ search_best_goal()
+ check_collisions()
+ rewire_goal()
+ choose_parent()
+ extend_closest_node()
+ smooth()
Kinematics
+joint_coordinates <double> vector
+joint_angles <double> vector
+ fk_solver(joint_angles) <double> vector
+ ik_solver(joint_coordinates) <double> vector
Control_System
- kp <double>
- ki <double>
- kd <double>
- sampling_time <double>
+ current_theta <double>
+ target_theta <double>
+ compute_PID(current_theta,target_theta) <double>
Map
+ x_max <double>
+ x_min <double>
+ y_max<double>
+ y_min<double>
+ z_max <double>
+ z_min <double>
+list_obstacles <double> vector
+ set_x_y_z_max_imits(double x_max,double y_max,double z_max)
+ set_x_y_z_min_limits(double x_min,double y_min,double z_min)
+ generate_obstacle(list_obstacles) <double>
Simulator


Start
Input Robot Configurations - Kinematics,Start and goal values
Input Kinematics,joint constraints,Joint space values
Input the Input Joint values to IK solver to return the goal Joint variable
We give the Input and Goal joint variables to RRT* Path Planner
Initalize the Planner
Generate Map with the considering the obstacles.
Start from the initial position with a parent node
Generate a random point around the space
Extend the parent node to the closest random point
Check for goal path
False
True
if the goal bias is less than the limit
Check for path_to_goal
True
False
Re-wire the nodes back to the start
Update the Path
Visualize the trajectorries for every timestamp
Input all the path variables to get a trajectory and simulate it through a simulator
Check for obstacles
True
False
Iterate the process till you reach the goal
generate a random point and then extend the previous node to a new closest node
Check for Collision
True
False