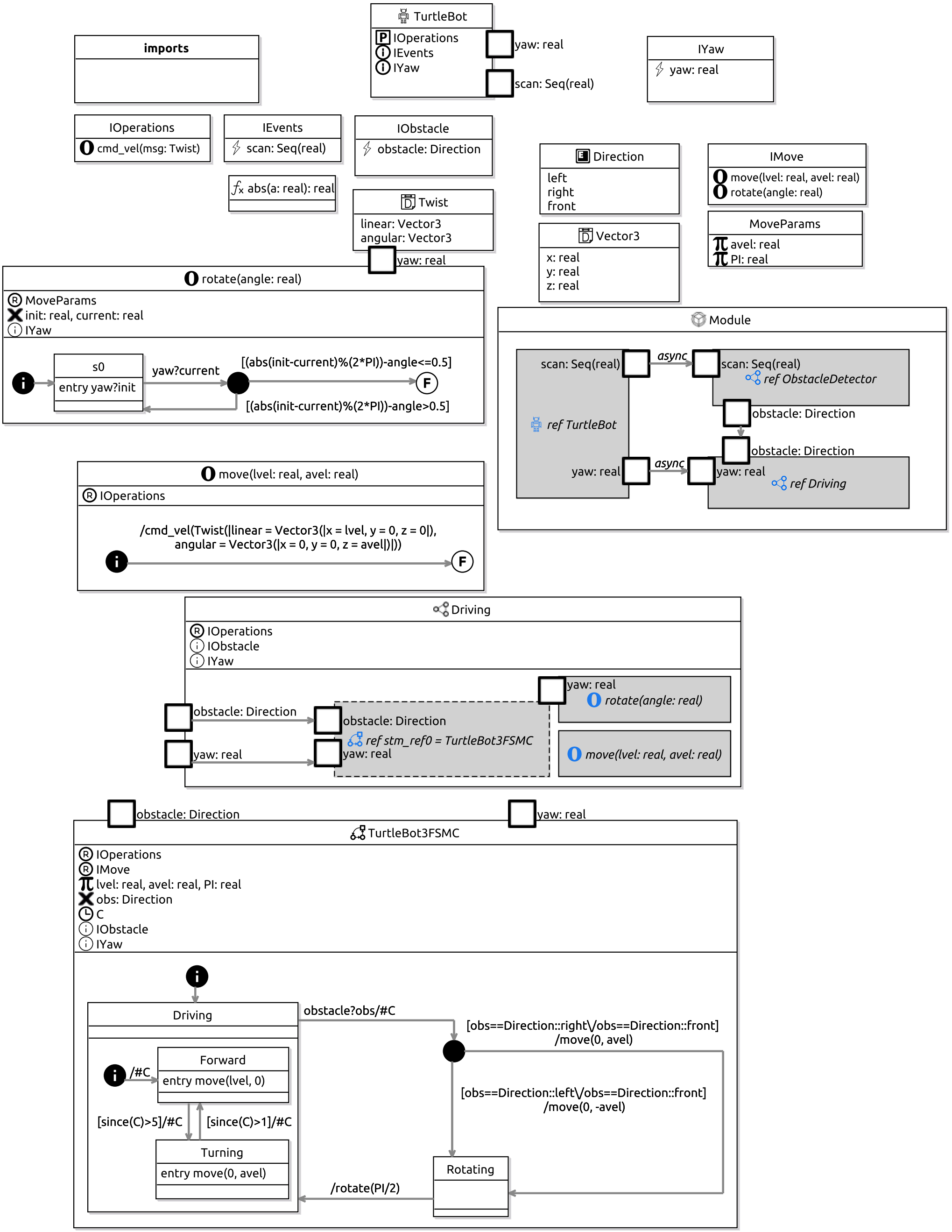
imports
TurtleBot
IOperations
IEvents
IYaw
yaw: real
scan: Seq(real)
IYaw
yaw: real
IOperations
cmd_vel(msg: Twist)
IEvents
scan: Seq(real)
IObstacle
obstacle: Direction
abs(a: real): real
Twist
linear: Vector3
angular: Vector3
yaw: real
Direction
left
right
front
Vector3
x: real
y: real
z: real
IMove
move(lvel: real, avel: real)
rotate(angle: real)
MoveParams
avel: real
PI: real
rotate(angle: real)
MoveParams
init: real, current: real
IYaw
s0
entry yaw?init
yaw?current
[(abs(init-current)%(2*PI))-angle<=0.5]
[(abs(init-current)%(2*PI))-angle>0.5]
F
Module
scan: Seq(real)
ref TurtleBot
yaw: real
async
scan: Seq(real)
ref ObstacleDetector
obstacle: Direction
obstacle: Direction
yaw: real
ref Driving
move(lvel: real, avel: real)
IOperations
/cmd_vel(Twist(|linear = Vector3(|x = lvel, y = 0, z = 0|),
angular = Vector3(|x = 0, y = 0, z = avel|)|))
F
Driving
IOperations
IObstacle
IYaw
obstacle: Direction
yaw: real
obstacle: Direction
ref stm_ref0 = TurtleBot3FSMC
yaw: real
yaw: real
rotate(angle: real)
move(lvel: real, avel: real)
obstacle: Direction
yaw: real
TurtleBot3FSMC
IOperations
IMove
lvel: real, avel: real, PI: real
obs: Direction
C
IObstacle
IYaw
Driving
Forward
entry move(lvel, 0)
/#C
[since(C)>5]/#C
[since(C)>1]/#C
Turning
entry move(0, avel)
obstacle?obs/#C
[obs==Direction::right\/obs==Direction::front]
/move(0, avel)
[obs==Direction::left\/obs==Direction::front]
/move(0, -avel)
Rotating
/rotate(PI/2)