

```

diagram AggregationController

type real
interface AggregationIface {
    MoveClockwise(i: real, j:real) : void
    RotateClockwise(i: real) : void
    var linearSpeed : real
    var angularSpeed : real
    event seeWall
    event seeRobot
}

stm AggregationFSM {
    requires AggregationIface
    initial I
    state S1 {
        entry MoveClockwise(-10.88, -0.75)
    }
    state S2 {
        entry RotateClockwise(-5.02)
    }
    transition T1 {
        from I to S1
    }
    transition T2 {
        from S2 to S1
        trigger seeWall
    }
    transition T3 {
        from S1 to S2
        trigger seeRobot
    }
}

operation RotateClockwise(i:real) : void {
    precondition i < 0

    requires AggregationIface
    initial I
    final F
    state S {
        entry linearSpeed = 0; angularSpeed = i
    }

    transition T1 {
        from I
        to S
    }
}

```

## AggregationController.rct

```
    transition T2 {
        from S
        to F
    }
}

operation MoveClockwise(i: real, j:real) : void {
    precondition i != 0 /\ j < 0

    requires AggregationIface

    initial I
    final F
    state S {
        entry linearSpeed = i; angularSpeed = j
    }

    transition T1 {
        from I
        to S
    }

    transition T2 {
        from S
        to F
    }
}
```