

0.1 Module Requirements: Full Auto Mode

The Full Auto Mode module consists of all the control for the robot in full auto mode. In this mode the user has no control over the robot except for stop and start. Once the robot is placed next to the tilt it will start moving in a well defined manner to achieve a set distance away from the tilt.

This mode makes strong use of the IR Sensors module uses the following functions from it:

- **IR_Calculate** - Calculates distance to tilt and current IR state.
- **Get_S_IR_state** - this function returns the current state of the IR where: error = 0, too close = 1, too far = 3.

Refer to IR Sensors module for more information.

0.1.1 Functional Requirements

As this module is for a run mode it requires many other modules to be initialized and running before operation. Many of these modules overlap with ones required to use the IR Sensor module. They include:

- **PWM Setup** converts Timer 2 to PWM mode necessary to output PWM for motor operation.
- **Serial Communications** to receive input from controller change mode to Auto and to start and stop the motors.
- **IR Sensors** to utilize the readings from the IR sensor for control. This also includes any functional requirements necessary for the IR sensors module such as the ADC converter setup.

Inputs The Full Auto module takes the following inputs from the IR Sensors module:

- **IR state** - This is the state of IR system which is a number from 0 to 3, where 0 = error, 1 = too close, 2 = parallel, 3 = too far.
- **newstate** flag - this flag is set whenever the IR sensor changes state (ie. from too far to too close, or from too close to error).

Process In Full Auto mode the jousting robot changes direction whenever the *newstate* flag is set. This change in direction corresponds to the current *IR state* which indicates if the robot is too close to the tilt (1), too far away from the tilt (3), or there is an error with the IR sensor and it is out of range (0). This leads to three possible movements:

1. IR state = ERROR (0). The robot stops moving and remains stationary.

2. IR state = TOO_CLOSE (1). The robot turns at a set speed and angle away from the tilt to the right.
3. IR state = TOO_FAR (3). The robot turns at a set speed and angle away towards the tilt to the left.

Outputs This module outputs a PWM signal to each motor controlling the direction the jousting robot moves.

Timing The speed at which the robot can change direction is determined on the efficiency of the IR sensor to quickly and accurately measure the current distance away from the tilt. But the rate that the velocity of the motors can be physically changed is affected by the motor hardware itself and any control system for the motors.

0.1.2 Non-Function (Quality of Service) Requirements

Performance This Full Auto mode can only be run when at very low speeds or it crashes immediately into the tilt. However even at low speed the robot will eventually drive into the tilt and stop. The movement path of the robot in auto mode is shown in the figure below:

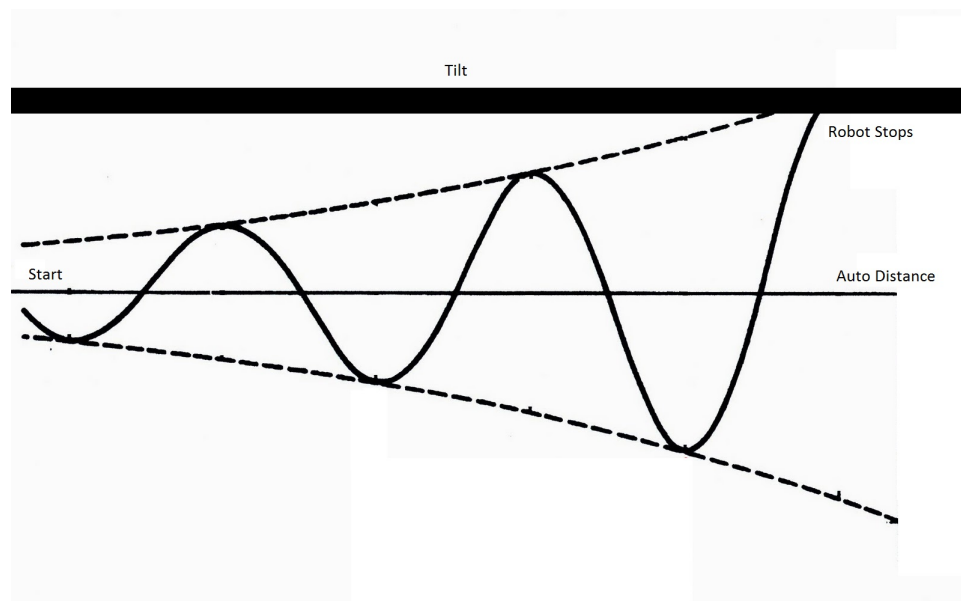


Figure 1: Full Auto movement path

Why this occurs is due to the angle of the IR sensor to the tilt increasing after each turn meaning that it cannot find the correct auto distance until it is

too late. However the main reason is thought to be the lack of responsive from the wheels when the *IR state* changes and the robot should turn in the other direction. There is a significant lag whenever the the motors change speed. The cause of this is the lack of feedback control for the motors.

Design Constraints As the motor control algorithm dd dot function successfully, there was a need to directly change the speed of the motor for each turn by hard-coding values directly into CCP1 and CCP2 to directly change the PWMs. However without any feedback from the motors this method is extremely unreliable and not very responsive.

0.2 Conceptual Design: Software Module

0.2.1 Design Rationale

The main focus of the Full Auto mode design was to make it simple. Without a working control algorithm for the motors it was decided a progressive rate of turning with relation to the distance calculated by the IR sensor would be difficult to implement. Therefore a fixed turn for both directions was implemented with the turning speeds tested by trial and error.

The reason why only one IR sensor was used is also for simplicity. The idea behind this method is that as long as the angle between the tilt and robot does not get too large there is only one distance where the robot is parallel to the tilt. This also requires starting the robot lose to this distance away from the tilt.

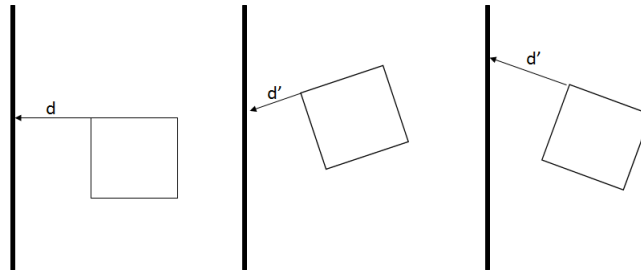


Figure 2: Full Auto IR distance

For this method to work effectively the angles of turn must be extremely shallow and the motors must be highly responsive and be able to change speed quickly.

0.2.2 Full Auto Flowchart

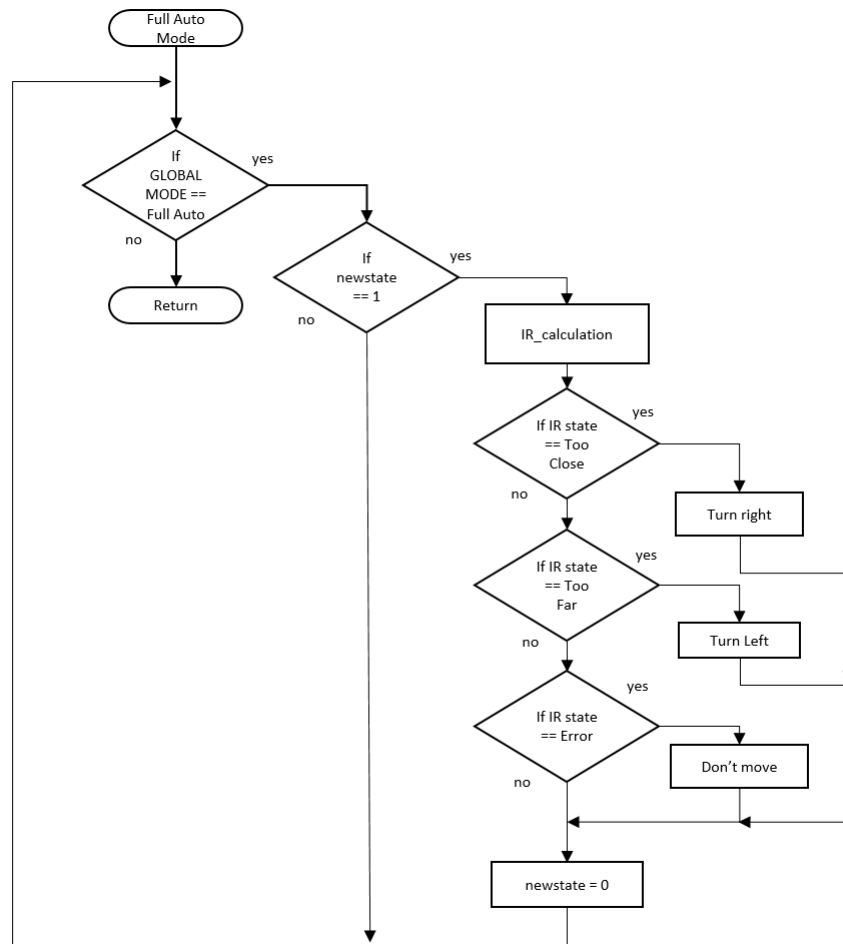


Figure 3: Full Auto Flowchart