Lab 0

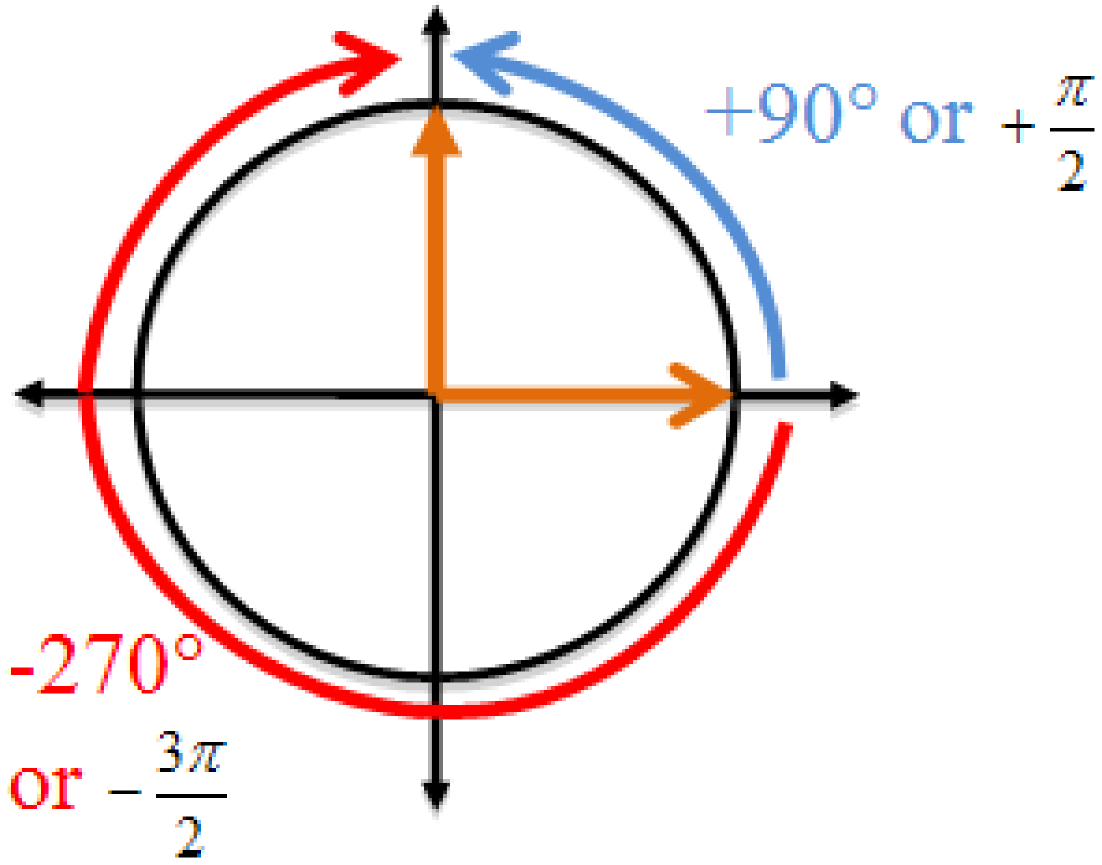
Robot Simulation

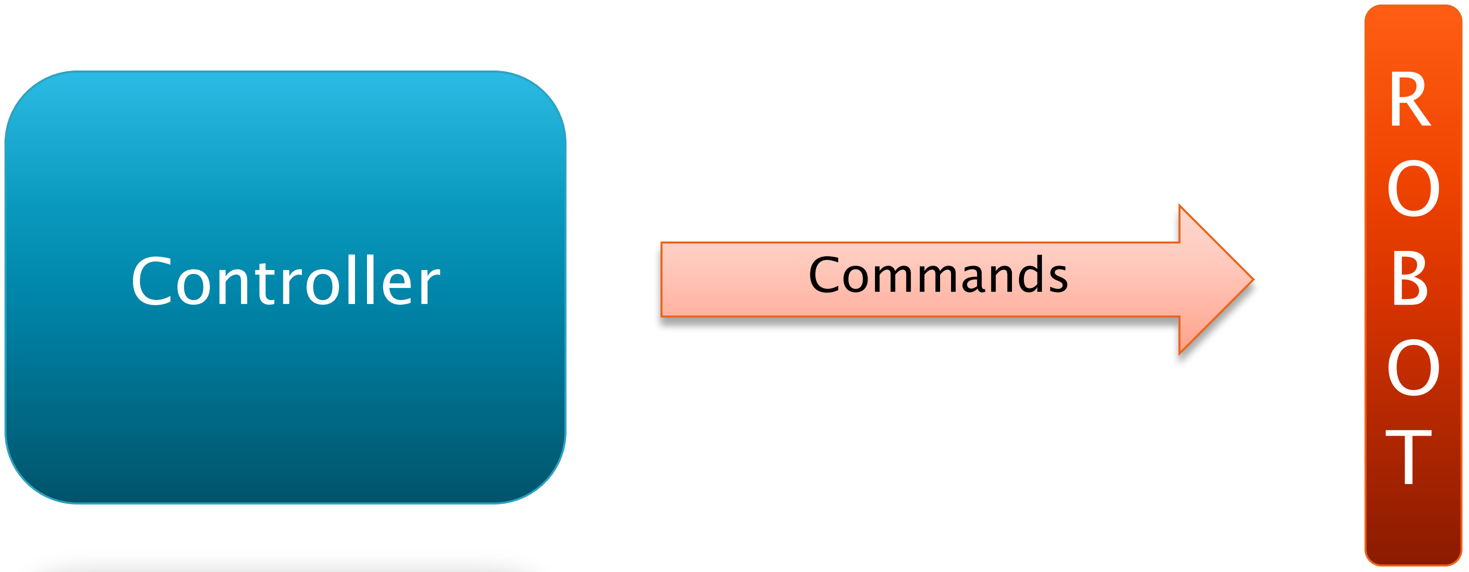
Purpose

- Use existing C++ skills to make calls to a rotating "Black-Box" robot. The implementation details are meant to be hidden except for the public interface.

- Use Object Oriented techniques to setup a "Controller" for the robot. The Controller is responsible for sending commands to the robot.

Rotate




Controller
Commands
ROBOT

Controller

- The Controller issues commands to the robot.
- The Controller may introduce helper classes to assist with the construction, transmission and cleanup of robot command messages.
- Think “Object Oriented Design”.

Commands

- A command is an 8-bit field layed out as:

OffOn	=	00000011
LoHi	=	00001100
Degree	=	11110000

Example Command

OpCode	ON	00000010
OpCode	HIGH	00001000
OpCode	180˚	00100000
Command		00101010

OP Codes

ZERO	=	00000000
OFF	=	00000001
ON	=	00000010
LOW	=	00000100
HIGH	=	00001000
90	=	00010000
180	=	00100000
270	=	01000000
360	=	10000000
EXEC	=	11111111

Public Interface

- The Robot has this interface:

  Constructor

  Default Constructor

  Destructor

  Default Constructor

  Execute(Commands&)

  Executes all the commands in the Command Queue.

Object Oriented Design

- This lab assignment could be easily written using C-style programming (i.e. procedural). The highest grade you can get for turning in a C-style assignment is a C.

- If you want higher than C (i.e. C++), use Object Oriented Design techniques.

- A sign you are using procedural is a lot of ‘IF’ statements.