HW 6 Algo Trading System

Due Date: Apri 8 in class

Problem 1 [Order Execution Module] The trading system we have discussed so far is overly simplified in executing orders. We basically just use one command line of IBMatlab to send market order to the exchange. Furthermore, this way of implementing the system basically assumes that the order will be executed without any problem. However that is not always true, especially in volatile market conditions. Hence, a good trading system should have a specialized function to handle the Order Execution. This problem gives you a good exercises in writing such a module.

The role of the Order Execution Modele is: once it is provided necessary info for trading, it will decide if it needs to split the transaction into a few small orders, and then send this series of small orders to the exchange one at a time. Hence the basic structure of the Order Execution Function is: it has inputs: Ticker, Share, Signal (buy/sell). Once these inputs are supplied to the function, the function decides if it should split Share into several smaller Shares. A primitive way of doing this is to divide the Share into fixed smaller shares, say 1,000. A better way is to use the current bid or ask size return by a price query. For example, made the smaller shares not too big (say not bigger than 1/3 of the bid/ask) compared to the current bid or ask size.

This problem is open ended, so use your creativity. For example, you can do the following:

(a). Use market orders: This can be done by generating fixed small size orders or random sized orders. In normal market conditions, and when the order size is not big, this usually works fine.

OR,

(b). Use limit orders: Limit orders gives you protection on the transaction price. One thing we need to note is the limit orders may not be executed immediately, so your function need to have a check on the execution of the order.

Note: Read the IBMatlab Guide about Order Execution, before you program this order execution Module/function.