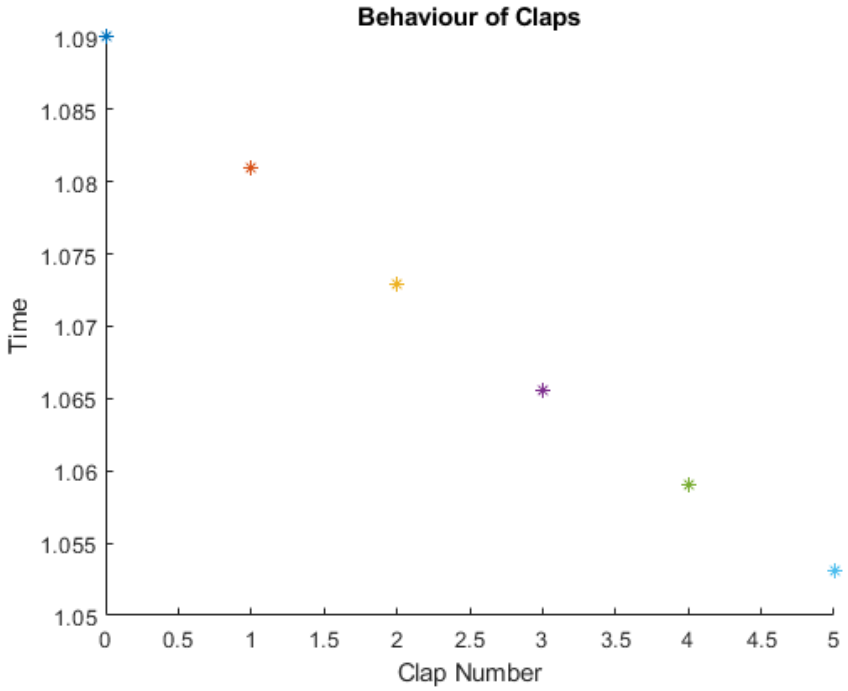
Synchronization

1. **Your first task is to execute labsync and understand what the clapping agent is doing out of the box. Set initialClapTime=.1. The agent will treat this initial clap time as an error because it is passed as error in the playClap function. The error passed is the difference between the actual clapping and the perfect metronome timing. Perform and experiment by changing the selfCorrection factor and observing what happens to the timing of the claps.**

   We performed the experiment using two values for selfCorrection 0.1 and 0.5. By changing the values of selfCorrection, we observed that the clapping agent adjusts its timing to maintain a consistent period despite the initial error. A higher value of 0.5 will lead to more rapid adjustments while the lower value of 0.1 will result in slower adjustments.

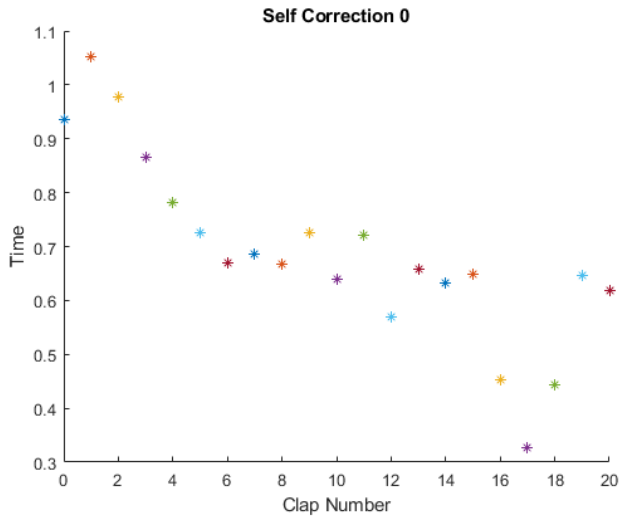
2. **Plot the six error values on a graph to obtain a graph akin to slide 18 in the lecture. Include the graph in your report and describe the behaviour of the agent relating it to the concept of metronome synchronisation.**

   The clapping agent starts with an initial clap time that is set as the error and as it claps repeatedly, the agent uses the selfCorrection factor to adjust its timing and reduce the error to synchronize with the metronome timing.

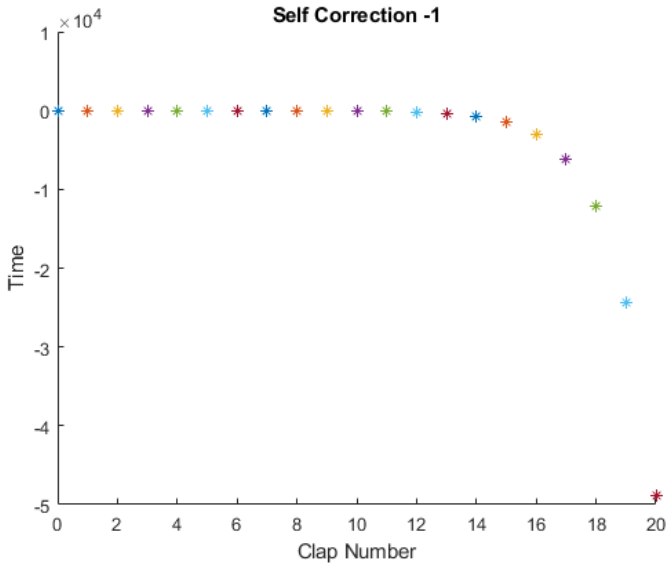


From the plot, we can observe that the initial error value is high and as the agent continues clapping it makes adjustments based on the error feedback, resulting in smaller error values. With each clap, the error decreases until it approaches zero, indicating synchronization with the metronome.

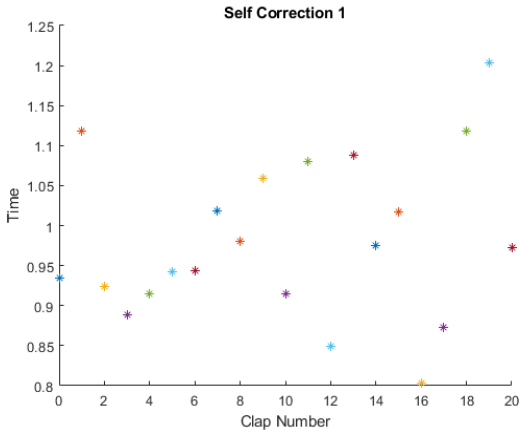
3. **Increase the number of claps to 20 by entering them in a loop and plot the errors as before. Set stdClappingError=.1 and initialClapTime=0. Observe how the behavior of the agent changes as a function of selfCorrection (e.g., use values of selfCorrection like -1, 0,1, 2). Execute the labsync script multiple times and plot graph lines on the same axes to test if a pattern emerges in the graph across different executions. Include the graph in your report. Describe the pattern found and write a comment on your findings about the change in agent behavior depending on the parameter value.**

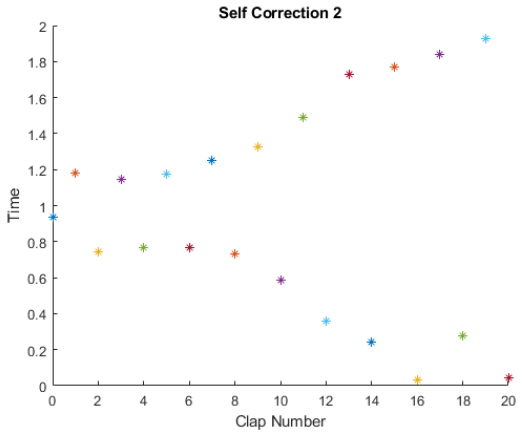


When the selfCorrection is set to 0, the plot shows a relatively stable pattern of error across claps and the agent adjusts its timing in a balanced manner to maintain a consistent period. The agent demonstrates moderate self correction to keep the clap timing errors within an acceptable range.



When the selfCorrection is set to negative value -1, the plot shows that the trend of the errors is increasing as the claps progress. The agent's timing adjustments are less effective in reducing the errors and the agent exhibits weaker self correction that results in a larger deviation from the metronome timing.

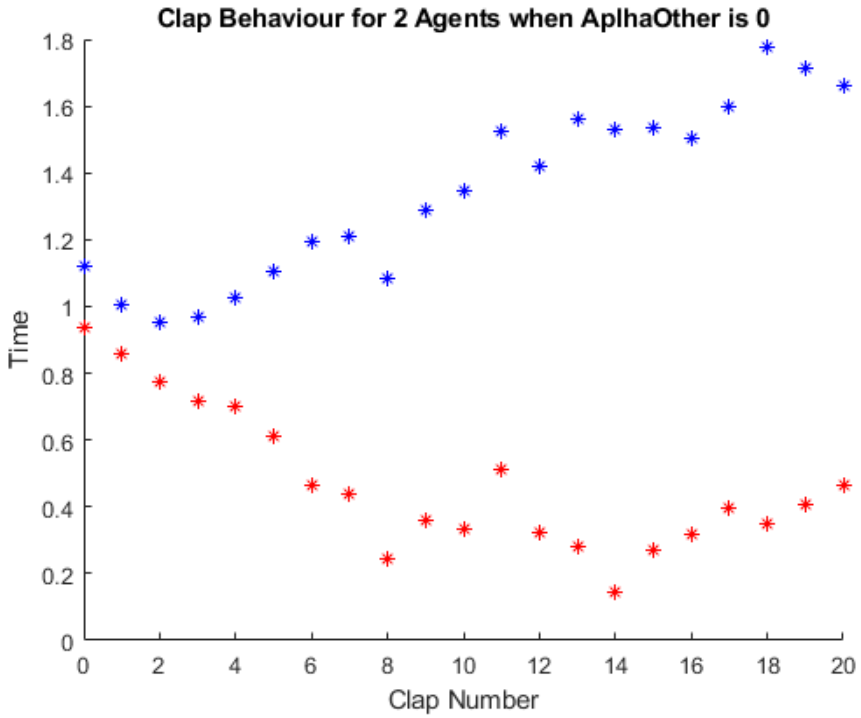




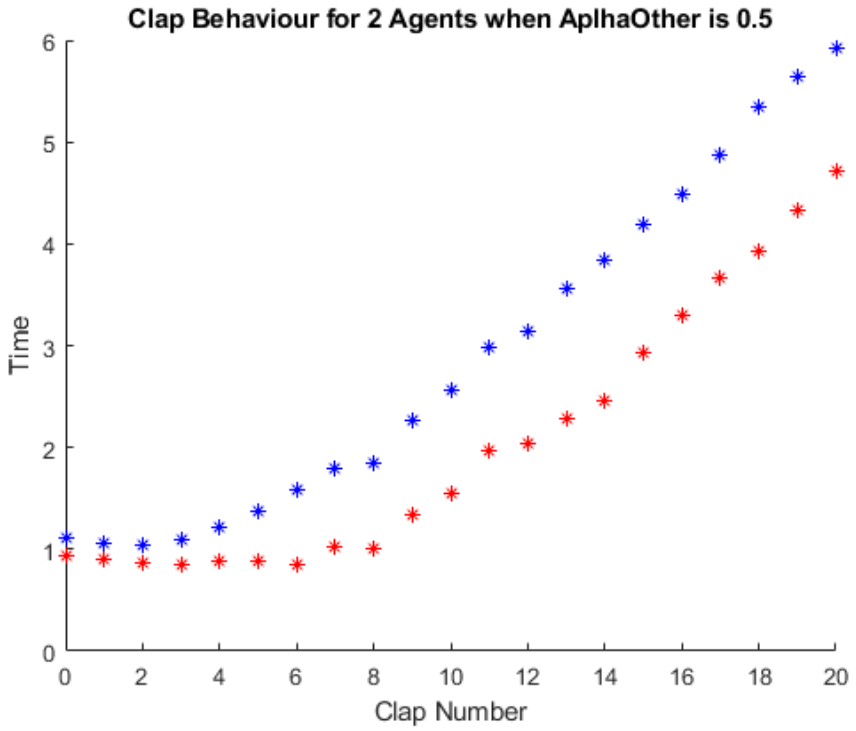
When the selfCorrection value is positive 1 and 2, the agents tend to adjust their timing more aggressively, resulting in faster convergence to the metronome timing. The agent exhibits a strong self correction mechanism that quickly minimizes the clap timing errors.

4. **Now we will look at how two agents sync to each other without an external metronome. Create two instances of clapping Agent that have correction values (alpha) with two variables (rather than one as in the previous task): [selfCorrection alphaOther]. By setting selfCorrection=0, the agent will ignore the metronome. The error passed to each agent will also be composed of two variables: the first is identical to the one with a single agent (but with selfCorrection=0 it will not be used), and the other is the asynchrony between two agents. Using a loop, show that the agents drift apart alphaOther=0, and maintain in sync with positive alphaOther. Using similar graphs as before, visualize one example of what happens with over-correction and one of what happens with under-correction and comment.**

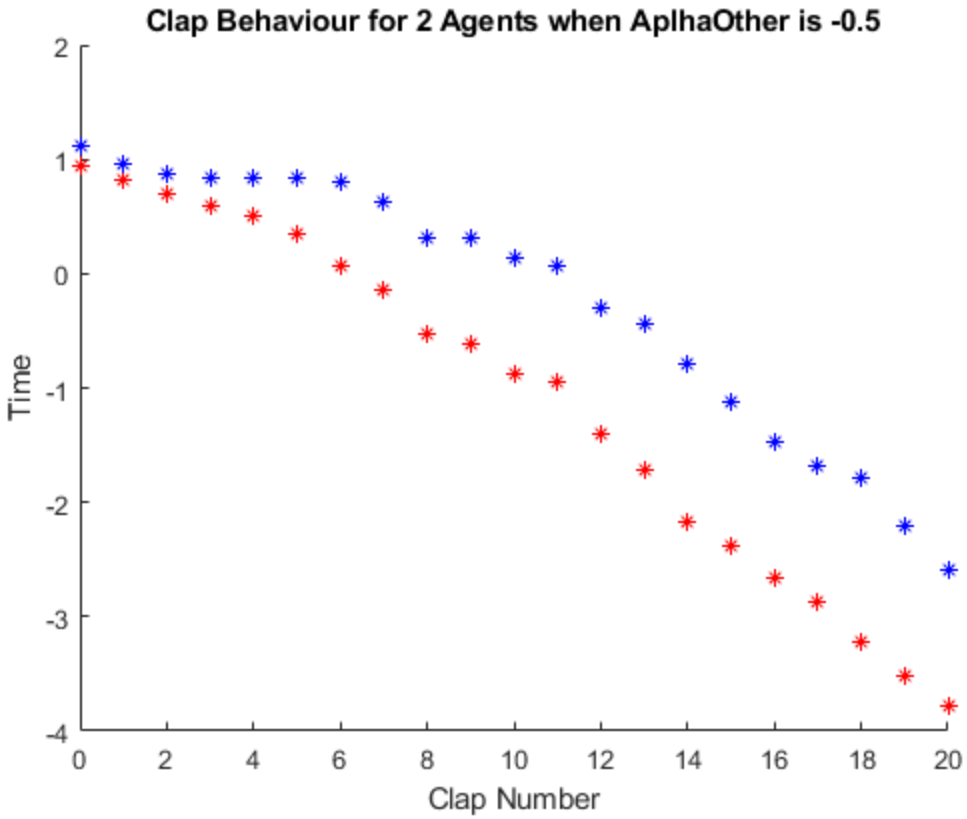
   When alphaOther is set to 0, There is no correction or interaction between the two agents based on the timings of each other's claps. The plot shows that the agents operate as individual entities, ignoring each other's timing and maintaining their own metronome-like behavior. As a result, there is no coordination or synchronization between them



For Overcorrection, we set the value of alphaOther as positive 0.5. The plot shows a decreasing trend in the errors as the claps progress and the agent makes aggressive adjustments to synchronize with each other. The agents tend to synchronize quickly and maintain a consistent period.



For Undercorrection, we set the value of alphaOther as negative -0.5. The plot shows an increasing trend in errors as the claps progress and the agent makes insufficient adjustments that result in gradual drifting apart. The agents struggle to maintain synchronization and gradually deviate from each other.


Clap Behaviour for 2 Agents when AplhaOther is -0.5
Time
Clap Number
2
1
0
-1
-2
-3
-4
0
2
4
6
8
10
12
14
16
18
20