Wireless Communication

Project 2

410786004 通訊四 蘇家駒

1. Abstract

Use matab to simulate RLS and LMS, then compare the difference between that two algorithms.

1. Introduction

In the family of linear filters computing their output according to

Our mission is to find the , but how to find it is a big problem. Here we use the recursion method to find it, as series of mathematical prove in the course, we get two method, RLS and LMS. Finding the recursively is very quick. And in this project, two algorithms I performed are below

RLS algorithm

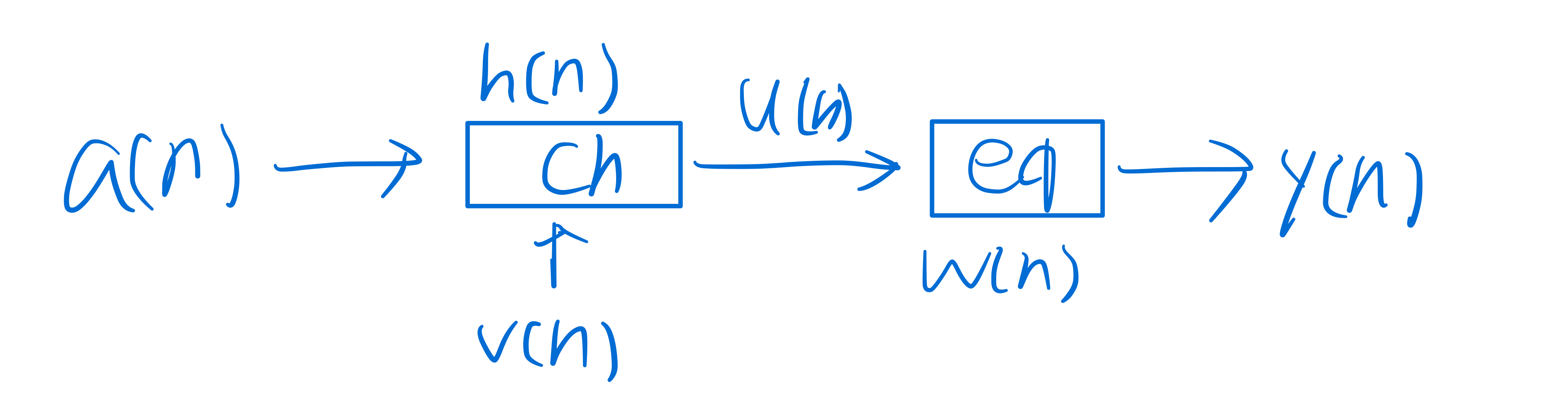
|  |
| --- |
|  |

LMS algorithm

|  |
| --- |
|  |

1. Methodology

In the beginning, I calculate size of all matrices I will use, and then simulate below system.



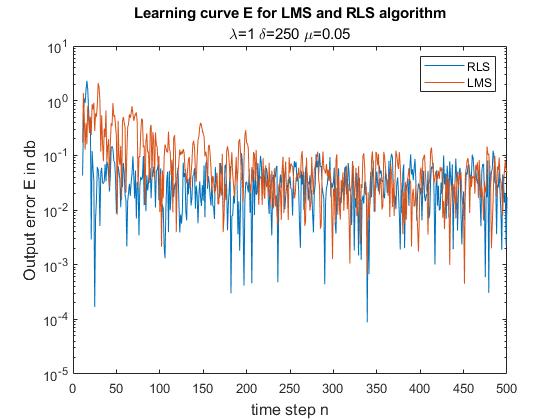
Simulated system diagram

In this case, is the source signal, put it into to channel and add noise then output

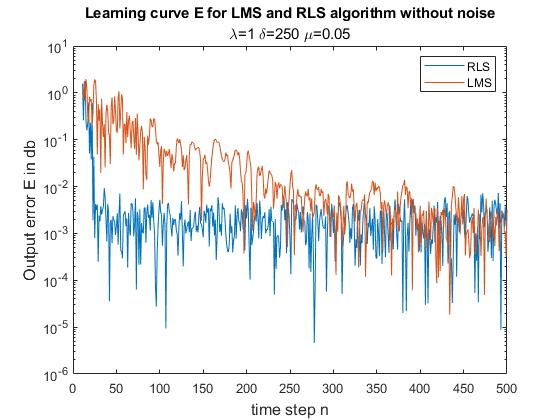
I performed two algorithms by writing function, pass the parameters I need and return the error back. In both of them, I first update and use this to get the , the signal I decode, and compare to , compute the error by . Finally, I plot each curve in different parameters and algorithms.

1. Conclusion

* Result
  + RLS vs LMS

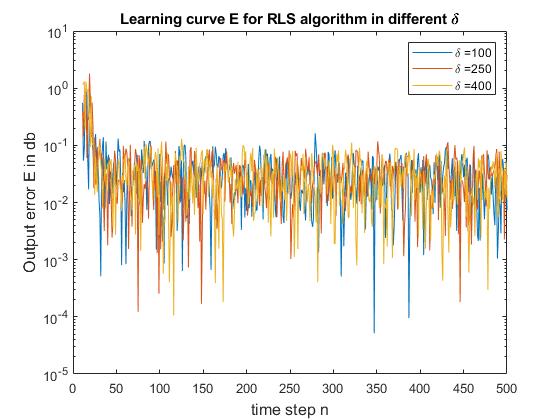


Above is the Learning curve for LMS and RLS, we can find the difference between two algorithm is that RLS converges faster than LMS, RLS quickly goes down in 25 step while LMS in 250 step. And I also simulate it again without noise and found the difference is clearer between them, the result as below



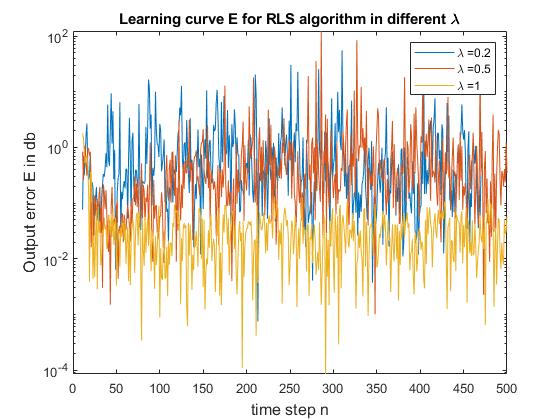
In communication system, although RLS has better performance than LMS, we usually use LMS as the adaptive filter because LMS is much easier to imply and the cost of LMS is lower than RLS, it is worth to sacrifice a little performance to get lower cost.

* + RLS in different



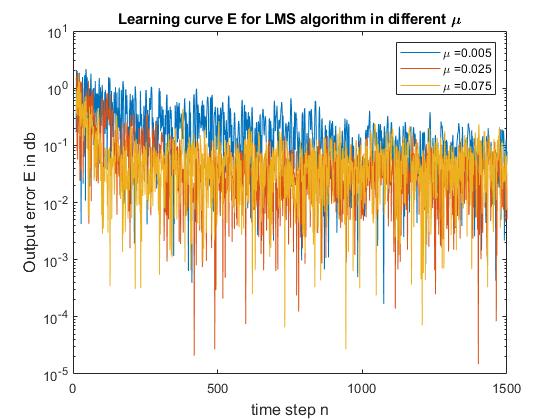
In course we say that , but I think is only relate to , so in the above figure it has the almost same curve because in this case, the .

* + RLS in different



Here is forgetting factor to control the importance of the relation between signal in different time, and it also control how quickly the learning curve converge. But I found when , the curve will converge first and then diverge.

* + LMS in different

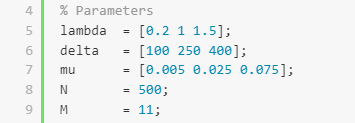


In general if is large then the curve converges quickly but get large error, otherwise, curve converges slowly and will get small error. And it react in the above result, compare with and , the former converges slower than the latter, but error is smaller.

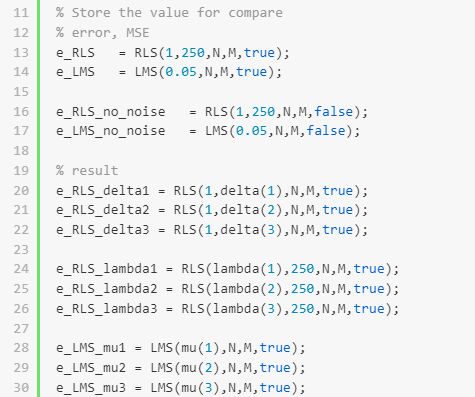
1. Code

My code separated in three part:

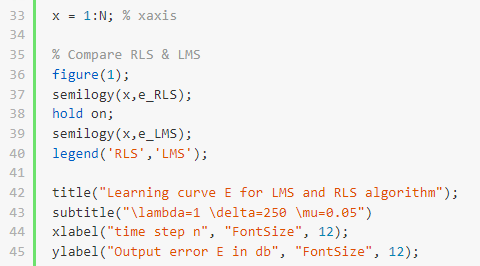
* + 1. project2.m: store error data and plot
    2. RLS.m: where the RLS algorithm implied
    3. LMS.m: where the LMS algorithm implied
* project2.m



Declare the constant I will use in that two algorithms.

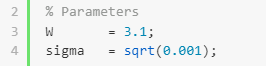


Declare and call function to store and compute the error with different parameter.

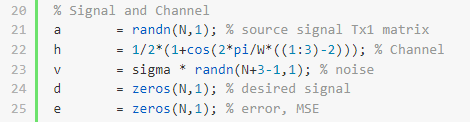


Plot the learning curve in different situations, above is similar to the other,

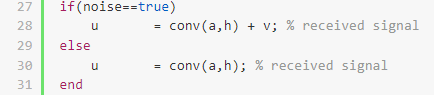
* RLS.m



Parameters used in source and noise.



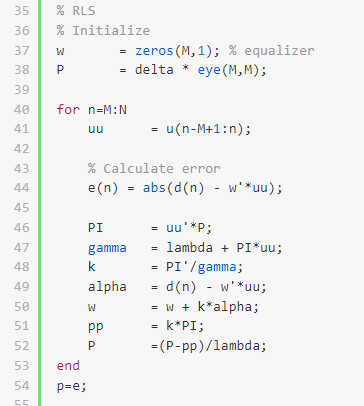
Signals and channel in the system.



Check if noise need to add.

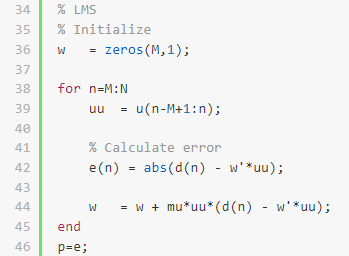


Perform



Perform RLS algorithm, first need to initialize the parameters, in each iteration, I calculate error first then update .

* LMS.m



The only difference to RLS.m is this part. RLS use MSE to find the best and LMS just use a constant to update its .