Sharif University of Technology
Electrical Engineering School

Advanced Neuroscience HW5
Based on Peter Dayan & Angela J Yu, 2002

MOTIVATION & CLASSICAL CONDITIONING

Armin Panjehpour
arminpp1379@gmail.com

Supervisor(s): Dr. Ghazizadeh
Sharif University, Tehran, Iran

08/05/2022

Part.1 - Rescola-Wagner(RW) Rule:

Using RW, we are going to design some of the well-known classical conditioning paradigms:

Paradigm	Pre-Train	Train	Result
Pavlovian		$s \to r$	$s \to ‘r’$
Extinction	$s \to r$	$s \to \cdot$	$s \to ‘\cdot’$
Partial		$s \to r \quad s \to \cdot$	$s \to \alpha\ ‘r’$
Blocking	$s_1 \to r$	$s_1 + s_2 \to r$	$s_1 \to ‘r’ \quad s_2 \to ‘\cdot’$
Inhibitory		$s_1 + s_2 \to \cdot \quad s_1 \to r$	$s_1 \to ‘r’ \quad s_2 \to -’r’$
Overshadow		$s_1 + s_2 \to r$	$s_1 \to \alpha_1 ‘r’ \quad s_2 \to \alpha_2 ‘r’$
Secondary	$s_1 \to r$	$s_2 \to s_1$	$s_2 \to ‘r’$

Figure 1: Paradigms

Part.1.1 - Pavlovian Conditioning

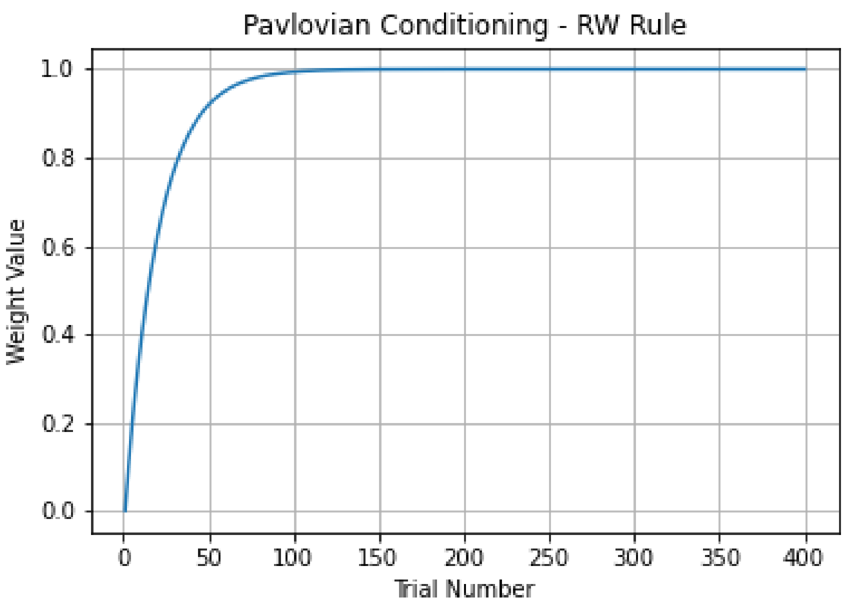


Figure 2: Pavlovian Conditioning

As we saw in the paper, the weight of the stimulus will increase to reach the reward value. (reward values is one here)

Part.1.2 - Extinction Conditioning

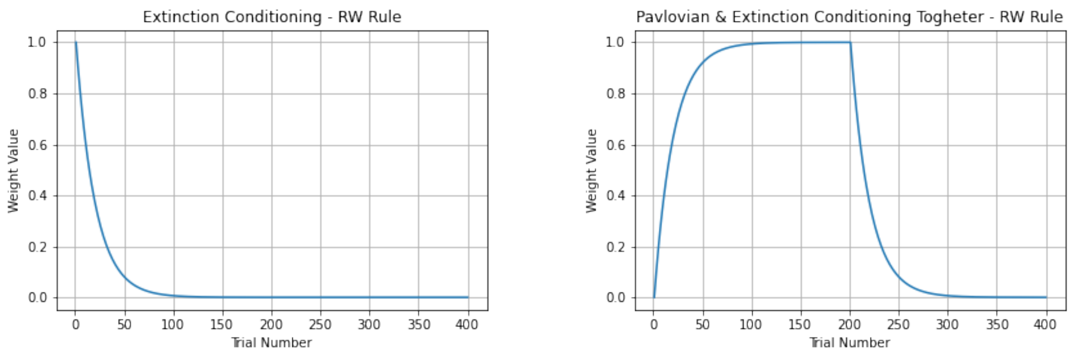


Figure 3: Extinction Conditioning

In the pre-training, the weight of the stimulus has been associated with the reward value. After that, in the training the reward was removed and as we can see, the weight of the stimulus is decreased to reach zero in training.

Part.1.3 - Partial Conditioning

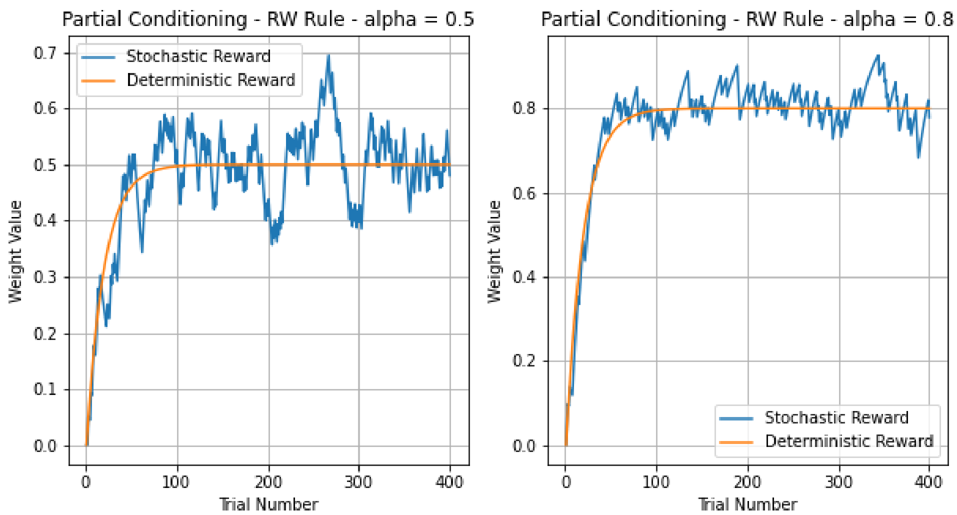


Figure 4: Partial Conditioning for α 0.5 and 0.8

As we expected, the stimulus reward will be increased and reach the probability of reward presentation.

Part.1.4 - Blocking Conditioning

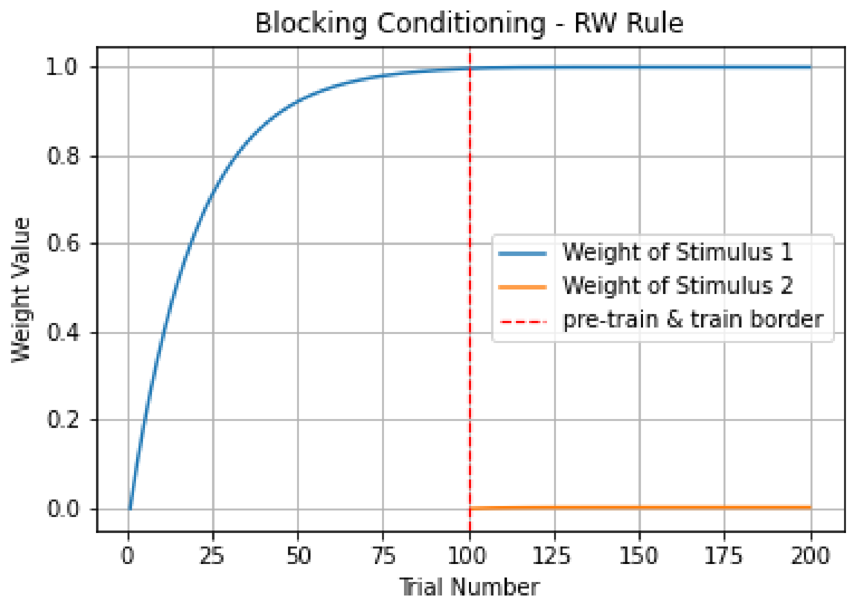


Figure 5: Blocking Conditioning

As we expected, the new stimulus's weight won't increase and stimulus 1 will have all the credits of the reward.

Part.1.5 - Inhibitory Conditioning

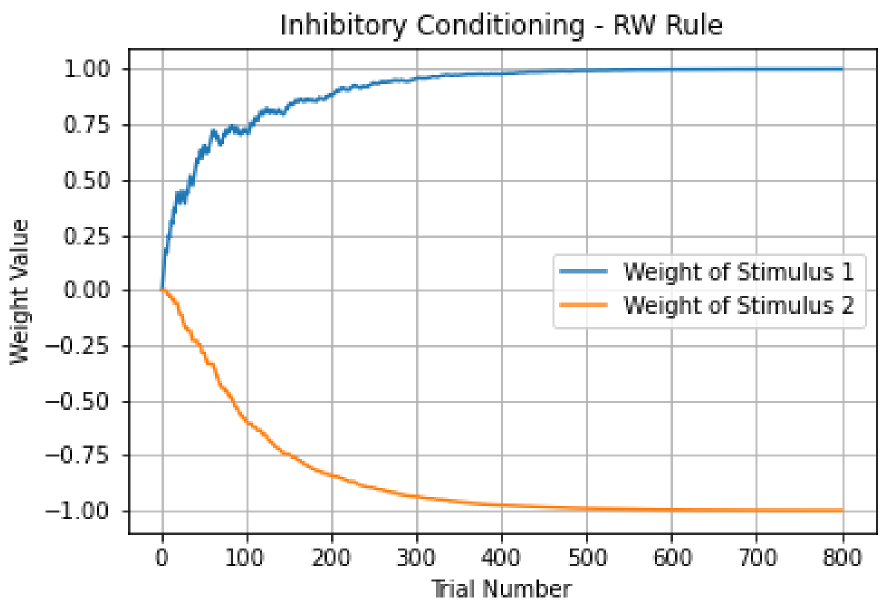


Figure 6: Inhibitory Conditioning

Same as the slides, one of the stimuli is an excitatory stimulus and the other is an inhibitory.

Part.1.6 - Overshadow Conditioning

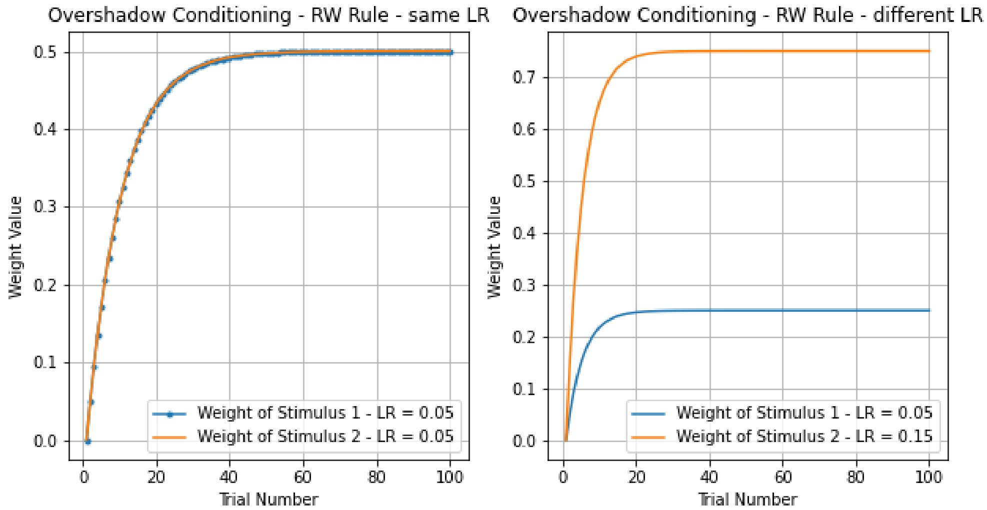


Figure 7: Overshadow Conditioning

As you can see, two stimuli with same learning rates, will have the same weights but if the learning rate of one is faster than the other, it will have more credit of the reward. No matter what, sum of the steady state weights of two stimuli is equal to the reward value. So with if the learning rates are different, the weights will have different amounts of learnings (weights).

Part.2 - Kalman Filter

Rescola-Wagner is not capable of implementing some more complex paradigms. In order to do that, we will use Kalman filter. In the first part we implement, blocking, unblocking, backward blocking, etc.

Part.2.1.1 - Blocking Conditioning

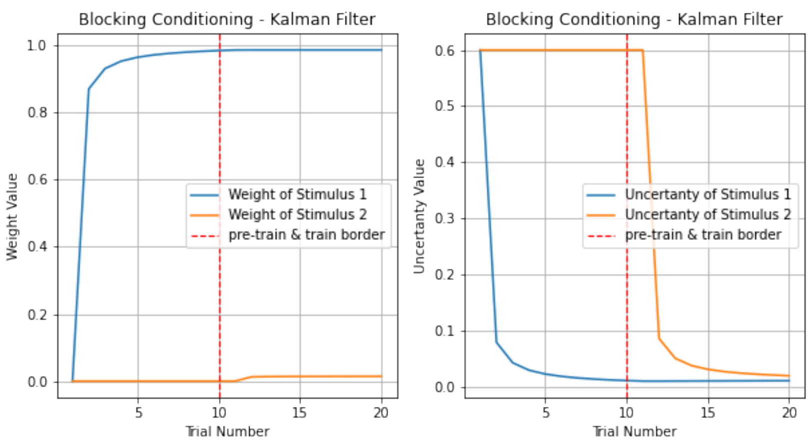


Figure 8: Blocking Conditioning - Weights and Uncertainty plots

As you can see, the pre trained stimulus has blocked the new stimulus. Uncertainties will decrease as time passes, too. The parameters of the Kalman filter can be found below:

τ	σ	σ_1	σ_2
0.6	0.01	0.6	0.6

Part.2.1.2 - Unblocking Conditioning

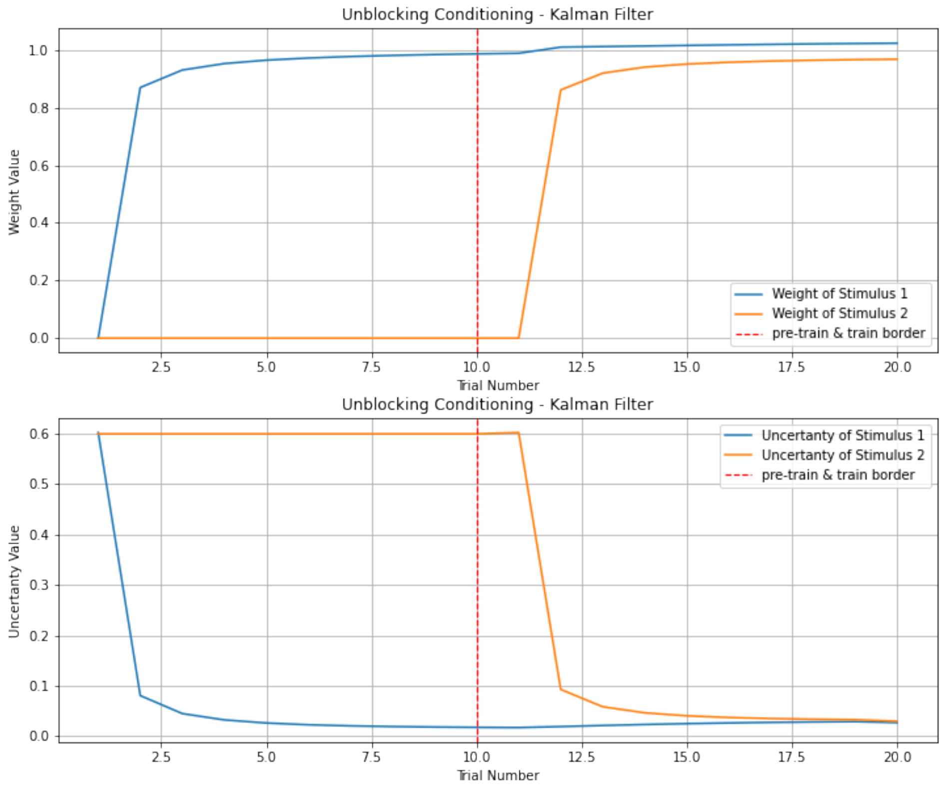


Figure 9: Unblocking Conditioning - Weights and Uncertainty plots

As you can see, at the start of the training, stimulus 1 has a low uncertainty so it won't change a lot, but stimulus 2 which has a high uncertainty will increase and get close to the weight of the stimulus 1. At the end, the sum of the weights is equal to the total reward in training which is 2r. The parameters of the Kalman filter can be found below:

τ	σ	σ_1	σ_2
0.6	0.05	0.6	0.6

Part.2.1.3 - Backward Blocking Conditioning

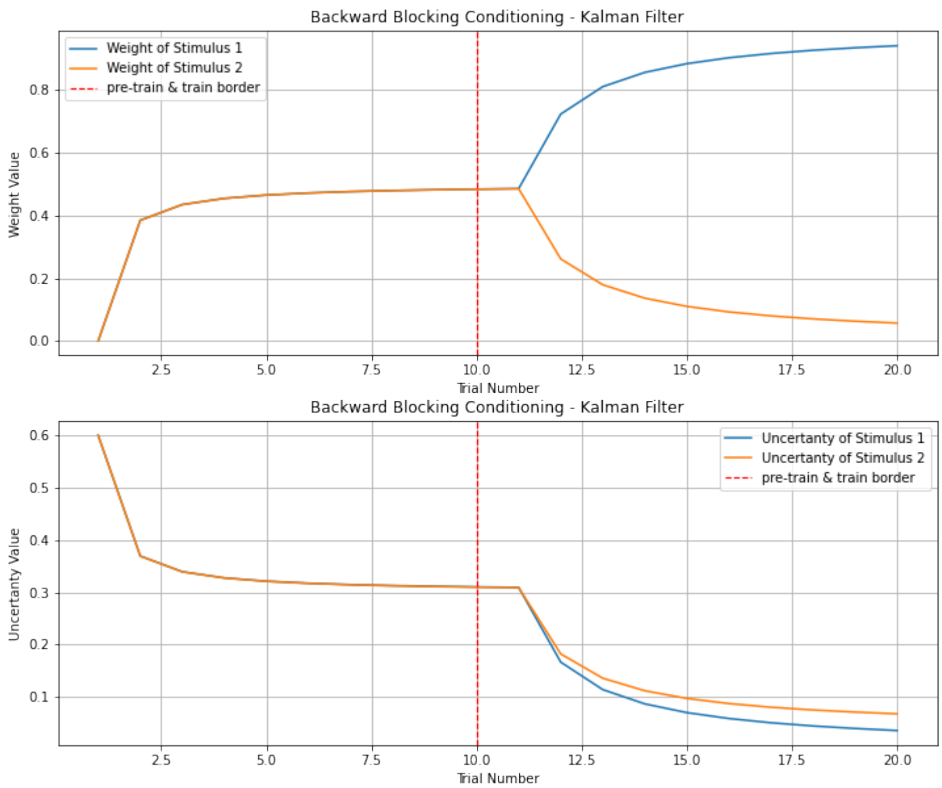


Figure 10: Backward Blocking Conditioning - Weights and Uncertainty plots

As you can see, after training, the subject realize that just stimulus one is responsible for the reward, so the all the credit of the reward goes to the stimulus one and stimulus two's weight will decrease to zero.

τ	σ	σ_1	σ_2
0.6	0.05	0.6	0.6

Part.2.1.4 - Backward Blocking Conditioning Joint Distribution of Weights

As you can see in the figure below, the joint distribution of the weights is plotted in three time samples, first, the 10th and the end time sample. As you can see, at first, the two weights are uncorrelated since their distribution forms a circle. at the second plot, the weights are anti-correlated as we saw at the first of the training in the figures of the previous part. The last plot shows that the mean of the w1 goes to one and the mean of the w2 goes to zero and the uncertainty of the w2 will have greater value. All of these results are matched with the result of part.2.3.

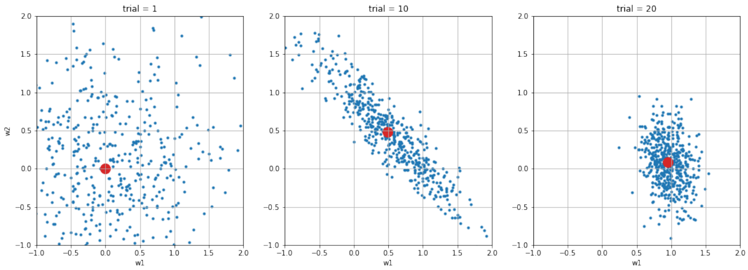


Figure 11: Backward Blocking Conditioning Joint Distribution of weights

Part.2.2 - Measurement and Process Noise Effect

Here, we simulate the results with three different values of measurement noise and process noise and here‘s the results:

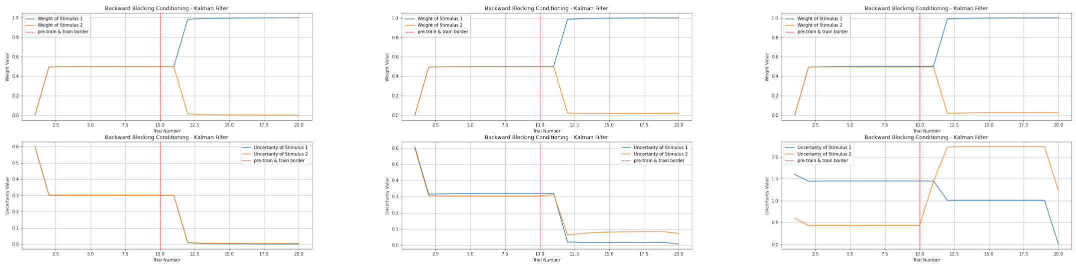


Figure 12: MNoise = 0.1, PNoise = 0.01, 0.1, 1

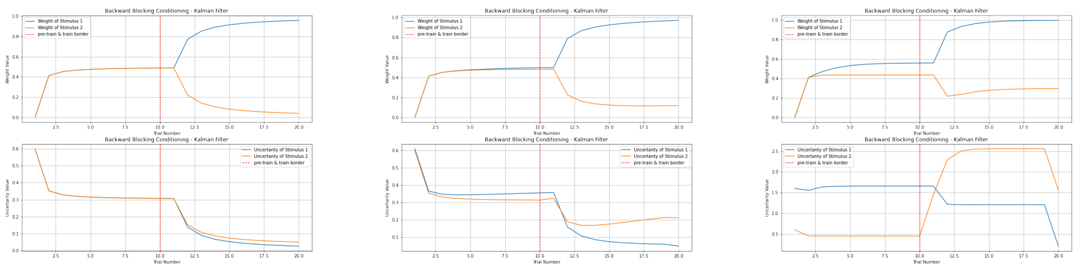


Figure 13: MNoise = 0.5, PNoise = 0.01, 0.1, 1

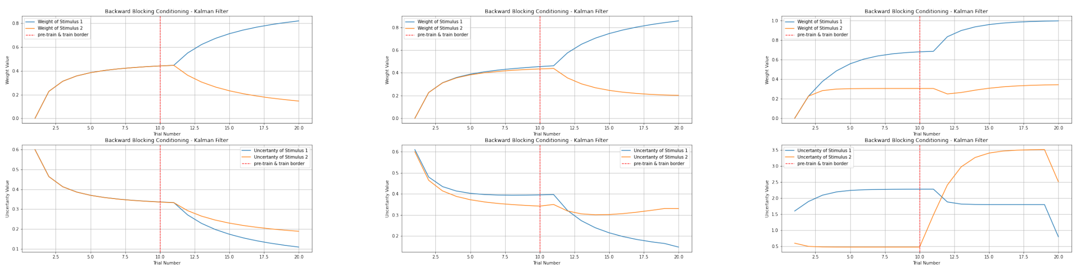


Figure 14: MNoise = 1.2, PNoise = 0.01, 0.1, 1

As you can see in the figures above, when we increase the measurement noise variance, it takes more time to get to the steady state and they change slowly since the environment is much more noisier. Also by increasing the process noise variance, the weights change faster because the covariance matrix as a coefficient will make bigger deltas.

Part.2.3 - Kalman Filter Steady State Gain

$$G_\infty = \Sigma_\infty^- C^T (C\Sigma_\infty^- C^T + V)^{-1}$$

$$\Sigma_t^- = A\Sigma_{t-1}^- A^T + W$$

$$\Sigma_t = \Sigma_t^- - G_t C \Sigma_t^-$$

Combining the three equations we have:

$$\Sigma_\infty = A\Sigma_\infty A^T + W - A\Sigma_\infty A^T C^T (C\Sigma_\infty C^T + V)^{-1}\Sigma_\infty$$

As you can see from the last equation, the steady state Kalman filter gain depends on the covariance matrix, A, C and the measurement noise.

Part.2.4 - Uncertainty Changes

As the equations tell us, the changes of the uncertainties which are the change of the covariance matrix doesn't depend on the error but it depends on the stimuli values and measurement noise.

Prediction:

$$\hat{x}_t^- = A\hat{x}_{t-1}$$
$$\Sigma_t^- = A\Sigma_{t-1}A^T + W$$

Update:

$$G_t = \Sigma_t^- C^T (C\Sigma_t^- C^T + V)^{-1}$$
$$\Sigma_t = \Sigma_t^- - G_t C \Sigma_t^-$$
$$\hat{x}_t = \hat{x}_t^- + G_t(y_t - C\hat{x}_t^-)$$

Figure 15: Kalman Filter Equations

Part.2.5 - Reward & Punishment Conditioning

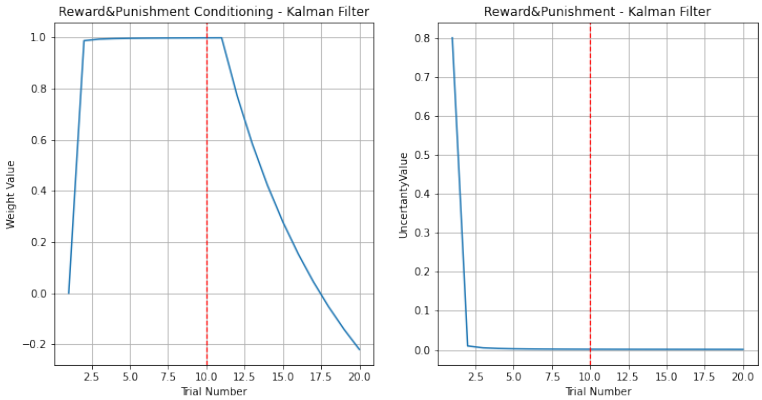
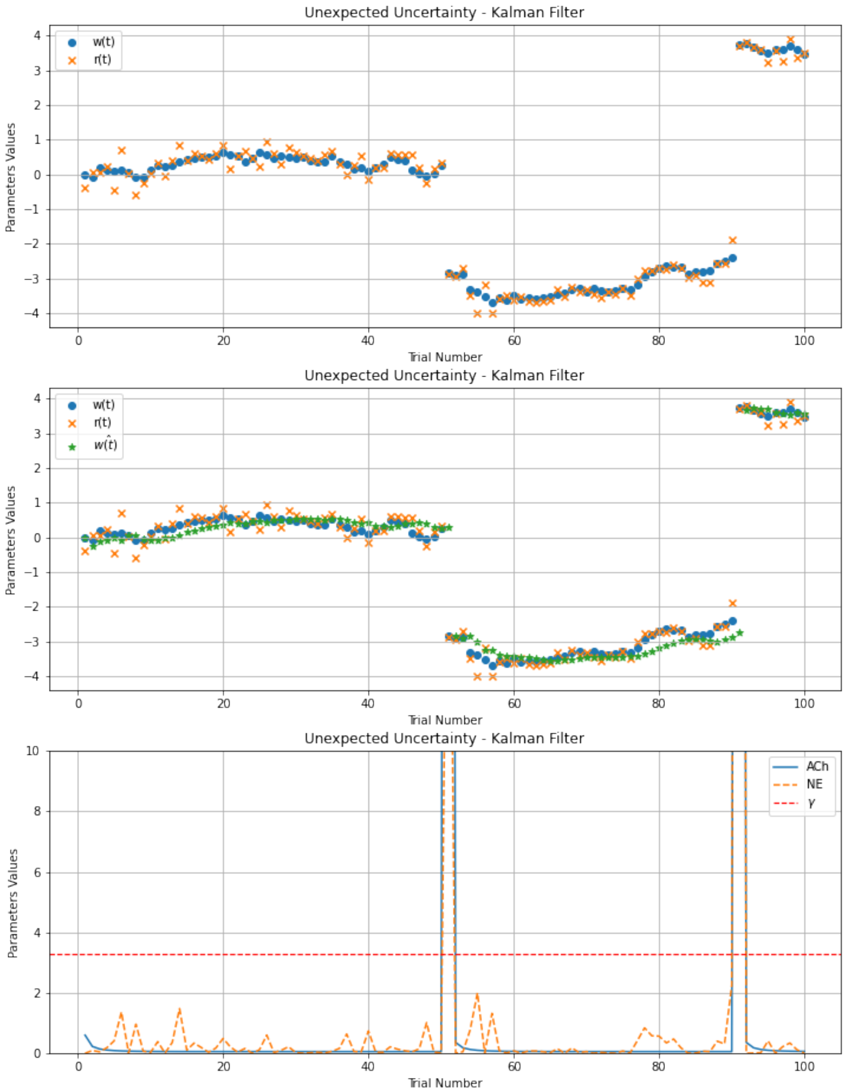


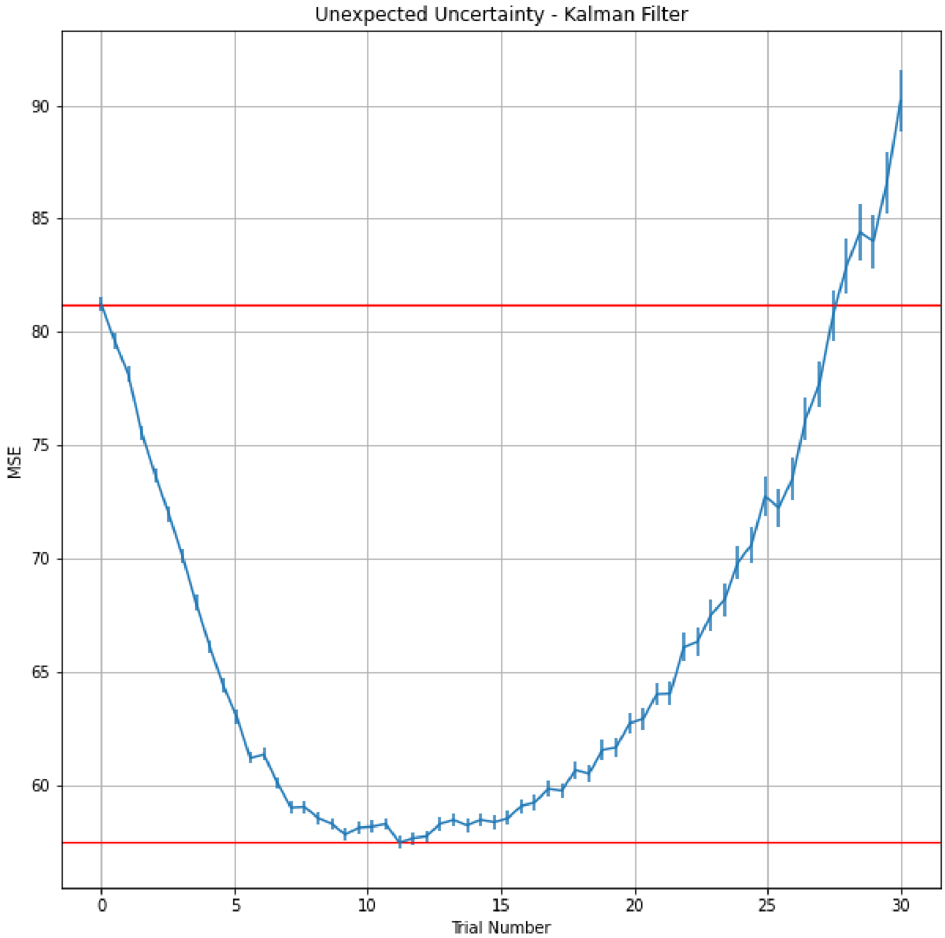
Figure 16: Kalman Filter Equations

As you can see, at the first, the weight will increase and reach the reward value so fast because at the first we have high uncertainty value but when the reward value changes to -r, the weight will decrease but not so fast because the value of uncertainty is already decreased. So this model can't notice the new changes in the environment, that's why we implement unexpected uncertainty model in the next part.

Part.3.1 - Unexpected Uncertainty

By measuring β(t) at each trial, we make sure we notice new changes in the environment. When sth changes, beta values increases and if it is passes a certain threshold, we increase the uncertainty values. Here‘s the result:





As you can see, when the threshold is too low or too high, MSE value is big and the optimum threshold seems to be around 10. In order to get a good figure, I run the code 1000 times and mean over the iterations for this last figure.