Active noise control without tap length selection: a model order weighting method

Yongjie Zhuang - *Stony Brook University*

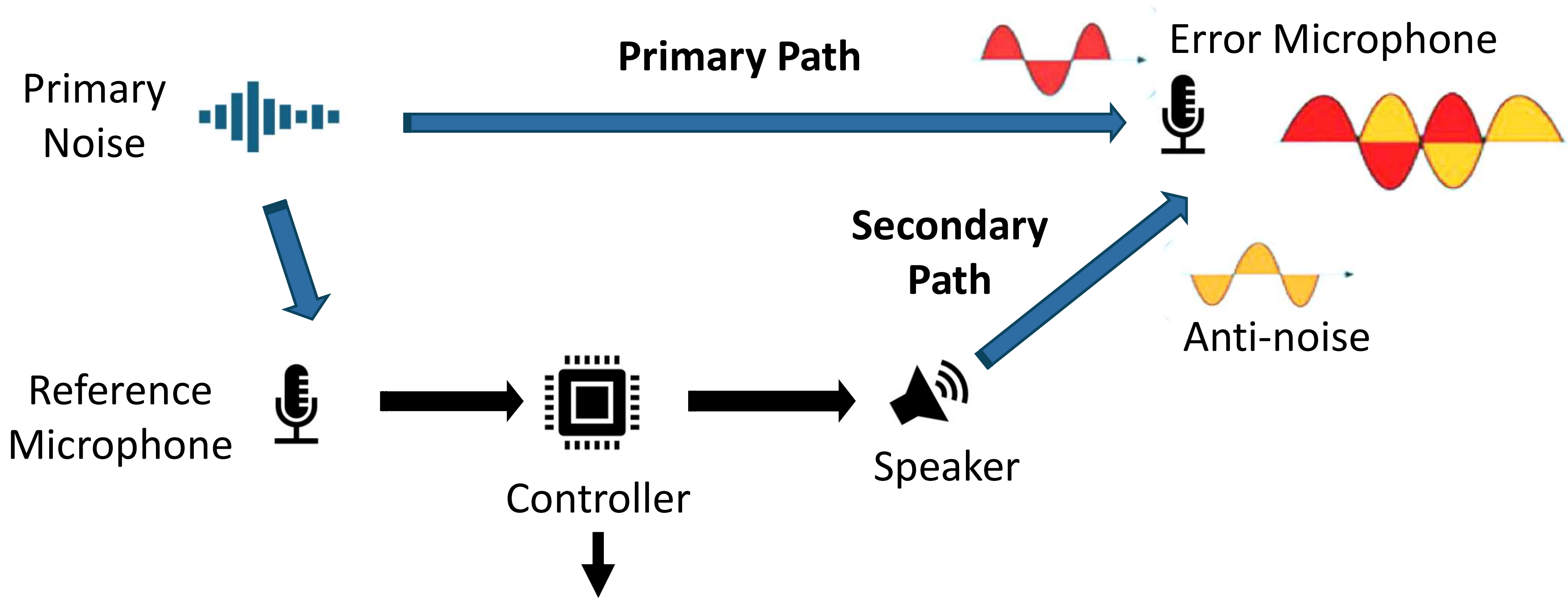
Manan Mittal - *Stony Brook University*

Rashen Fernando – *Discovery Partners Institute and University of Illinois Chicago*

Ryan M. Corey – *Discovery Partners Institute and University of Illinois Chicago*

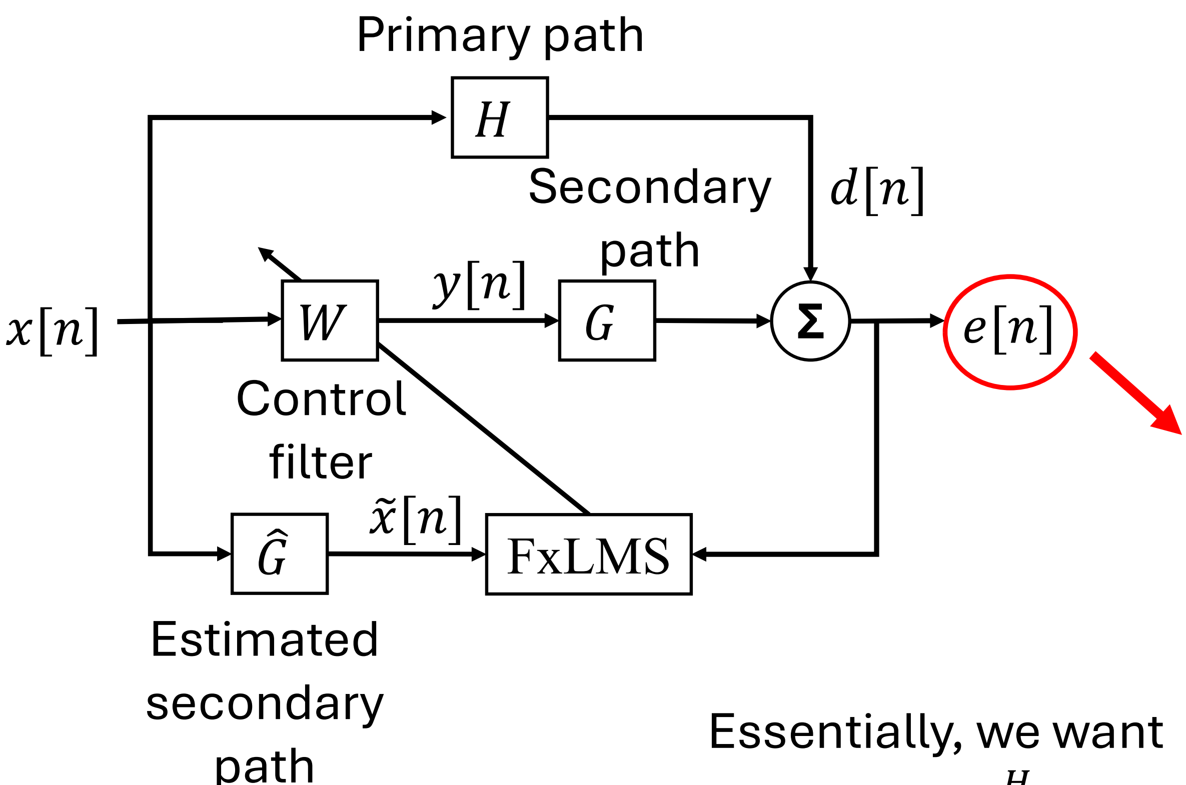
Andrew C. Singer - *Stony Brook University*

Active noise control (ANC) usually use fixed model order (tap length)



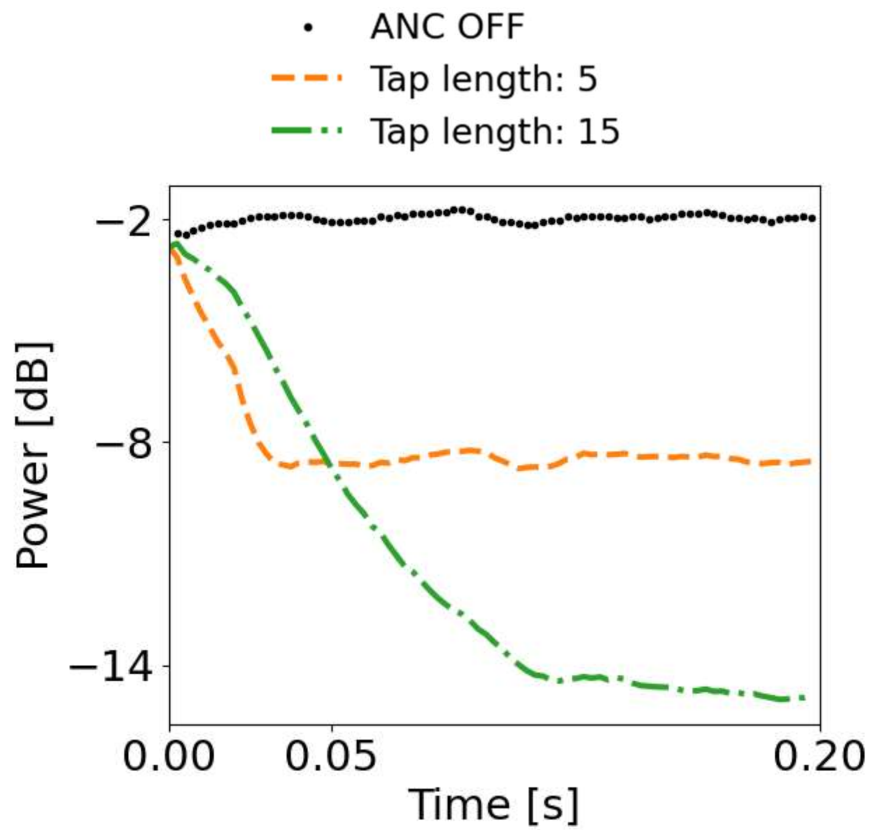
Usually, a fixed tap length finite impulse response (FIR) filter is used in the controller

Shorter tap length: faster convergence from eigenvalue analysis
Longer tap length: better steady-state performance

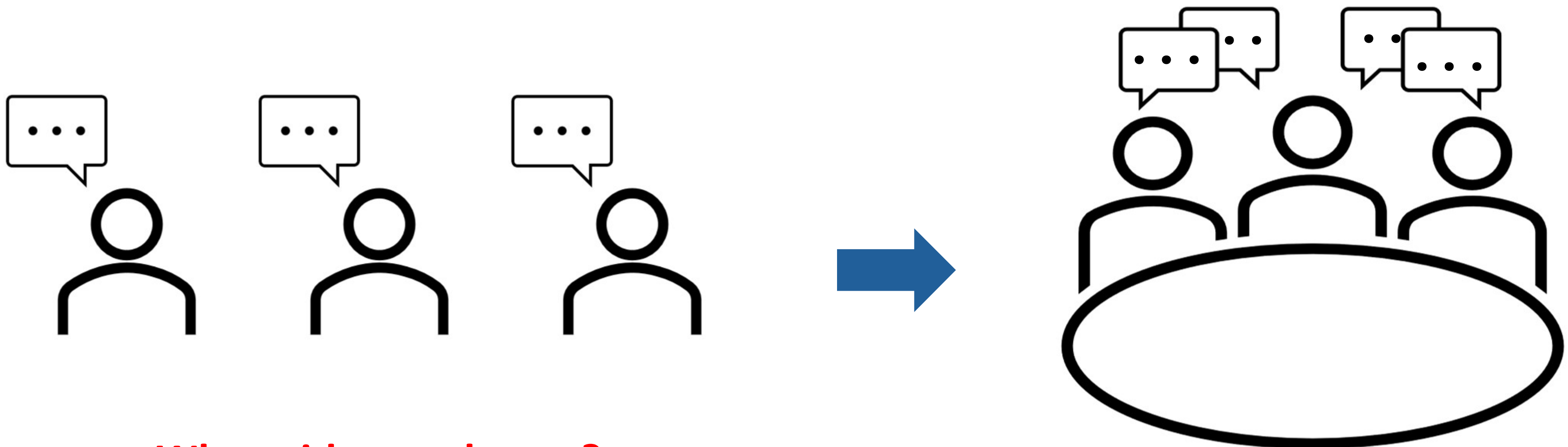


Essentially, we want

$$W = -\frac{H}{G}$$



Intuition on the proposed method: a mixture of experts



Whose idea to choose?

Blend their ideas based on their previous performance!

In real life:

- The trade-off between convergence rate and steady-state performance makes it difficult to determine the optimal tap length
- The optimal tap length may shift over time in time-varying environments

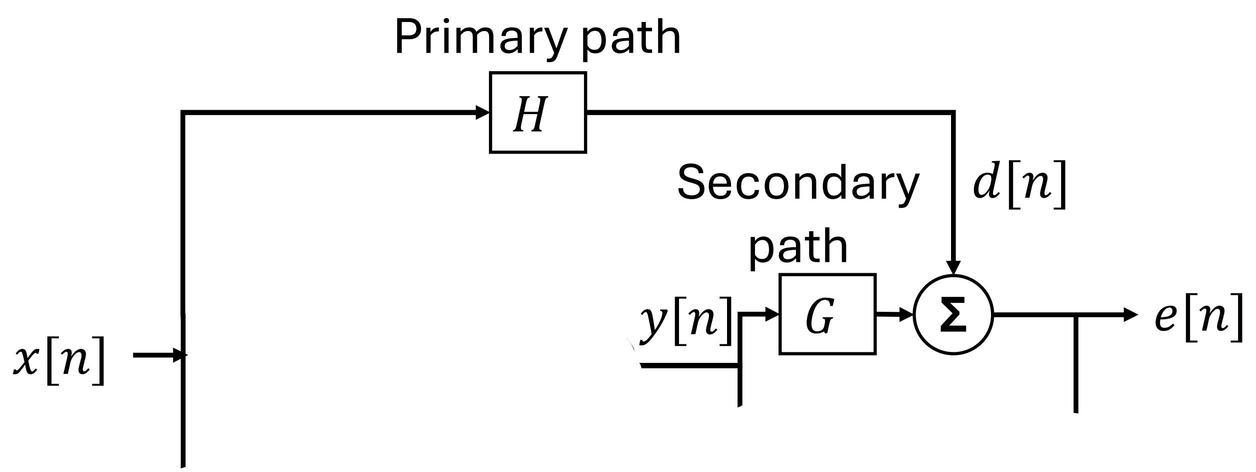
Instead of determining which model order to choose, blend different model orders.

In [Singer & Feder, 1999]:

- Linear predictors of different model orders are weighted to get the prediction.

- No need for tuning the model order.

- Their method is **universal**:
  Asymptotically achieving the performance of the best candidate predictor

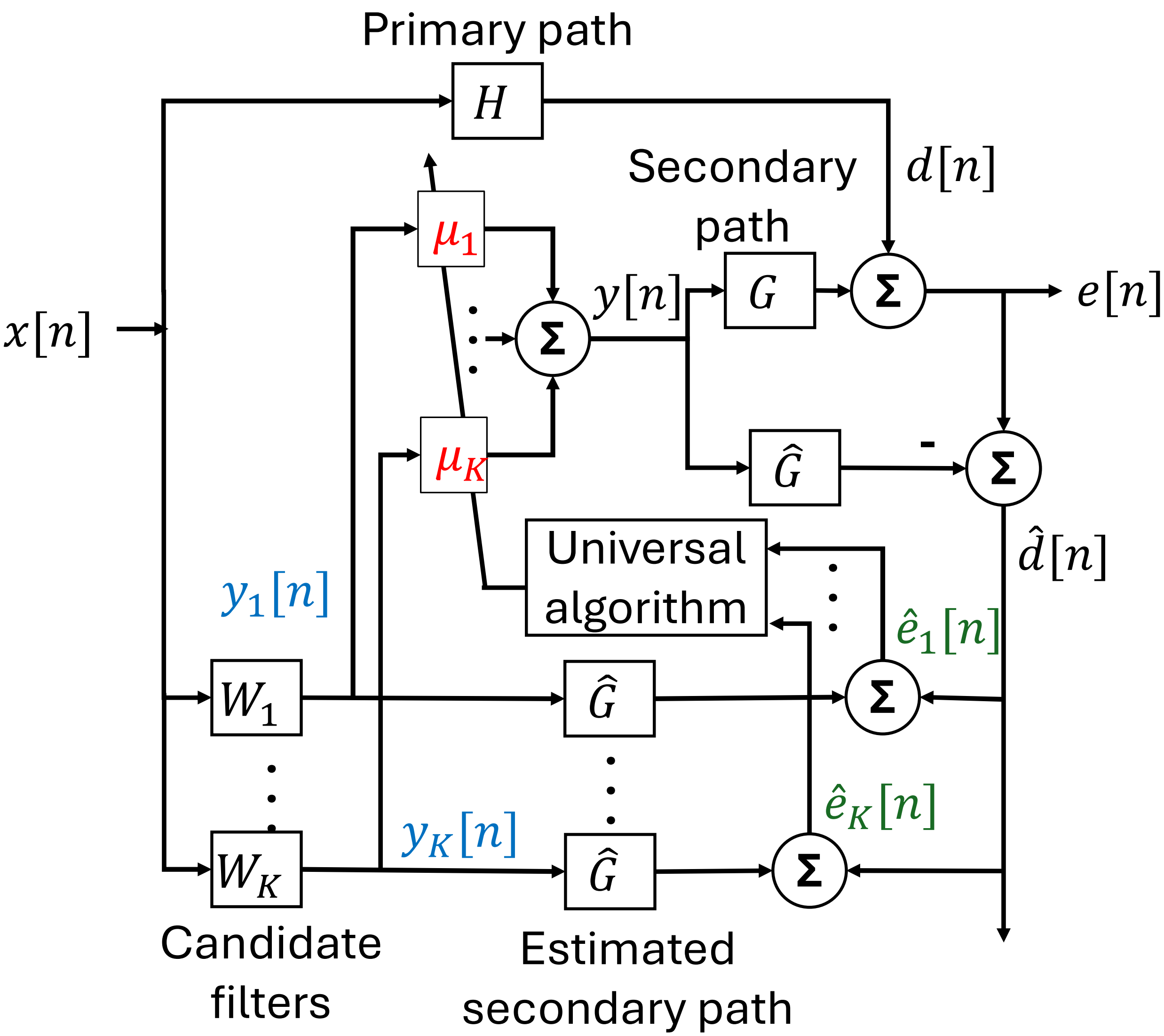


- The physical input of the speaker is:

Candidate filter outputs

$$y[n] = \sum_{k=1}^{K} \mu_k[n] y_k[n]$$

Mixture weights



- The mixture weights are computed based on each candidate filter's performance using softmax:

Performance of kth filter

$$\mu_k[n] = \frac{\mathrm{e}^{-\alpha \ell_{n-1,k}}}{\sum_{j=1}^{K} \mathrm{e}^{-\alpha \ell_{n-1,j}}}$$

Normalizing the weights

$\ell_{n-1,j}$: cumulative noise power at $\hat{e}_j[n-1]$ with forgetting factor

Universal method can track the best candidate filter in both transient and steady-state phase

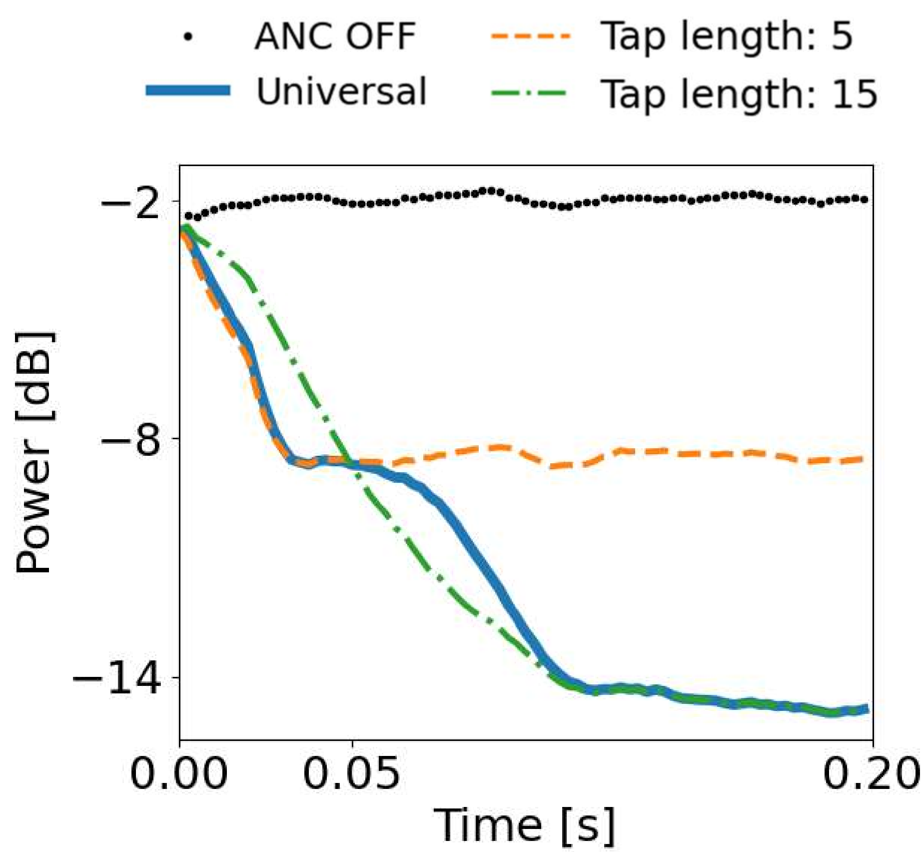
Apply the universal method on the previous example (100 Monte Carlo trials):

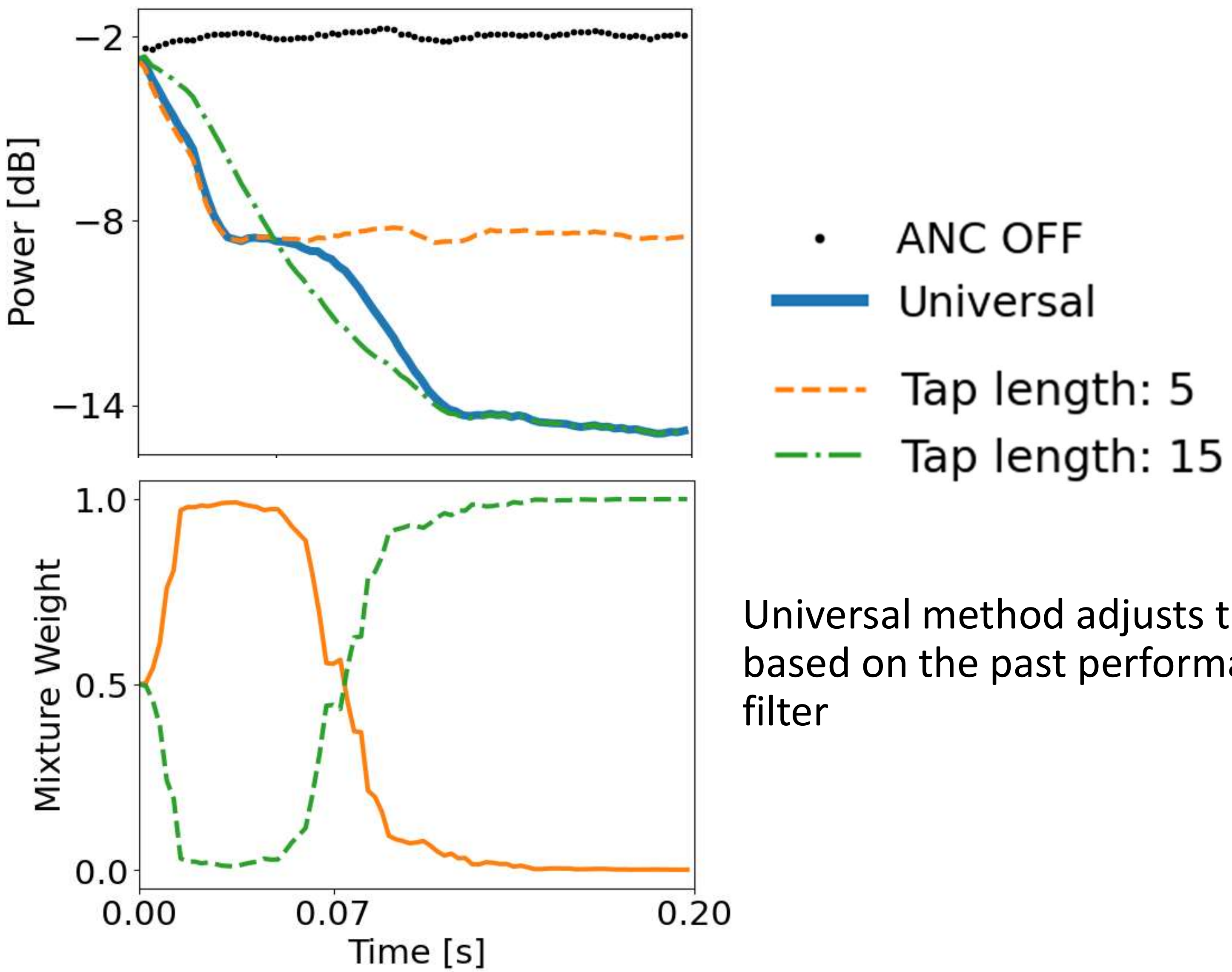
- Primary path:

$$H(z) = 1$$

- Secondary path:

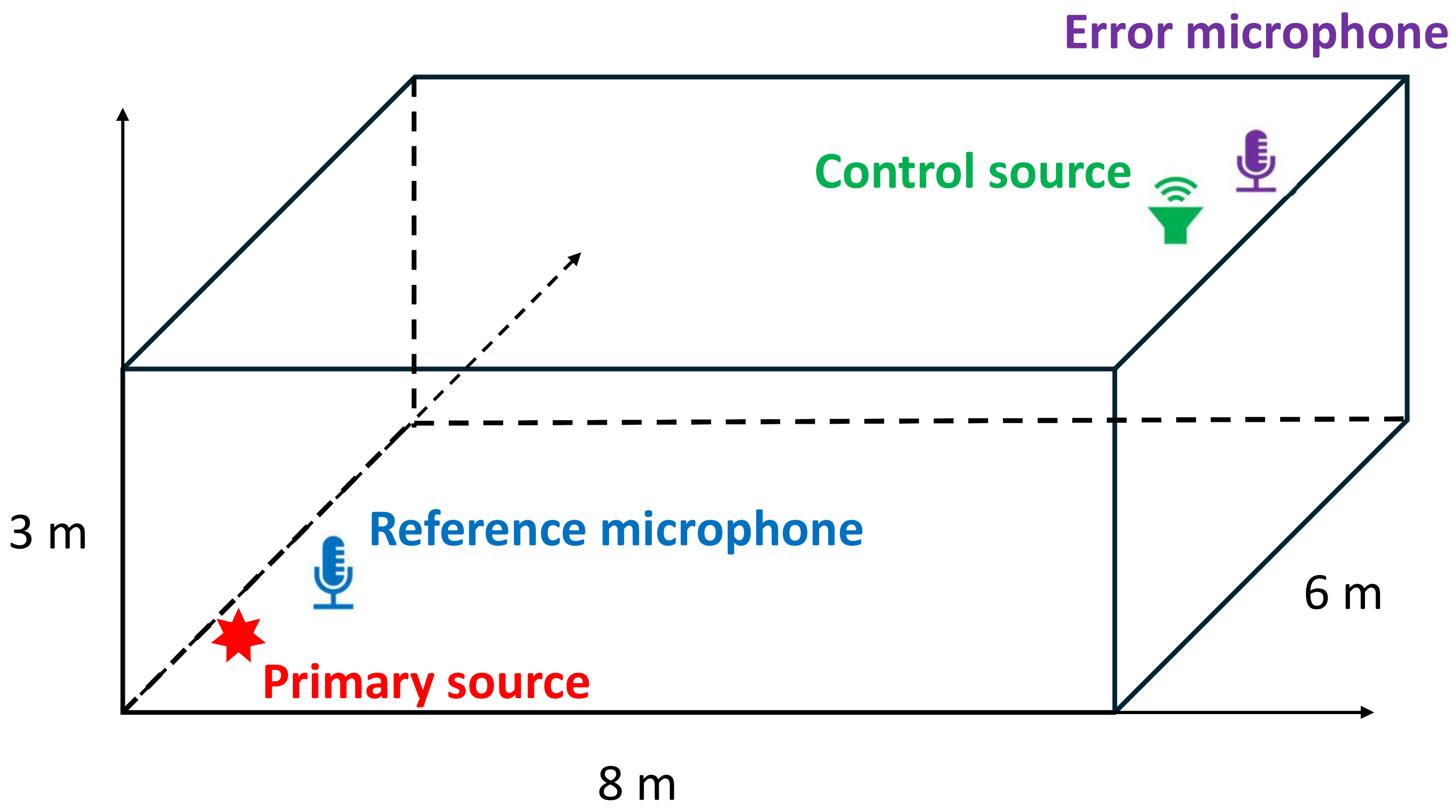
$$G(z) = 1 + 0.5z^{-1}$$



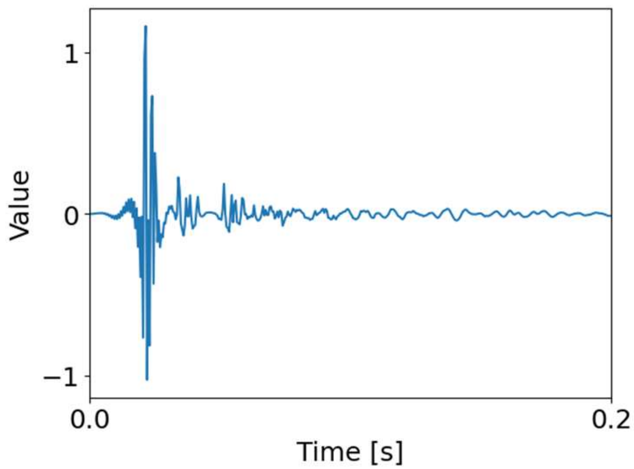


Universal method adjusts the mixture weights based on the past performance of each candidate filter

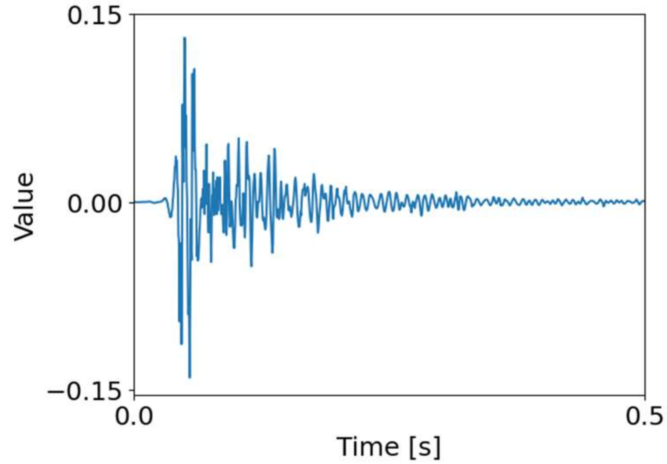
Using Pyroomacoustics to simulate a room response



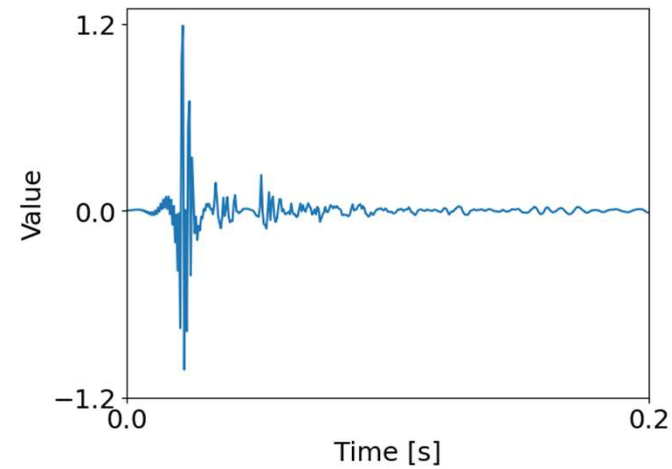
Simulated room responses



Noise source → Reference mic



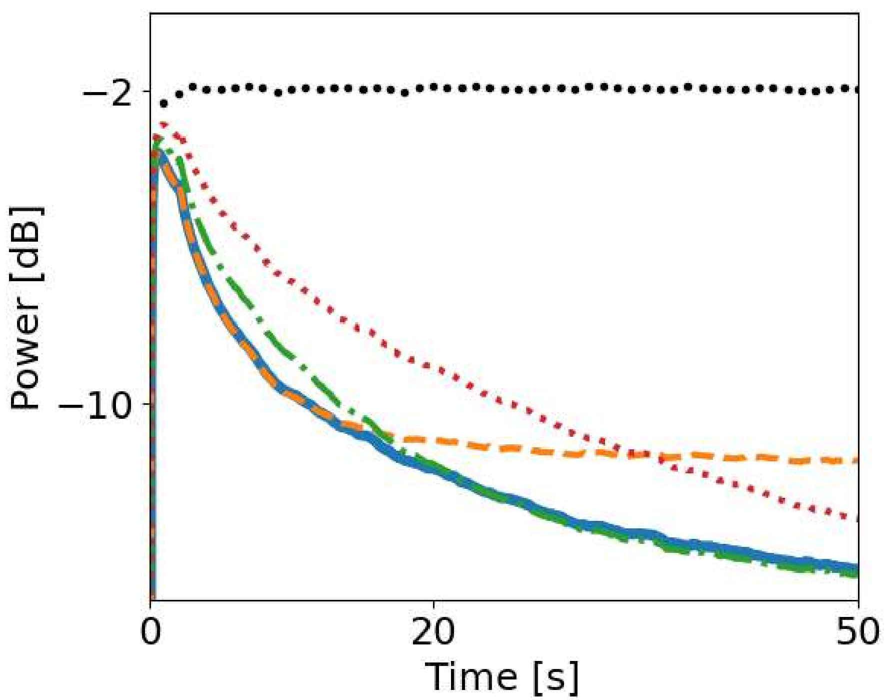
Noise source → Error mic

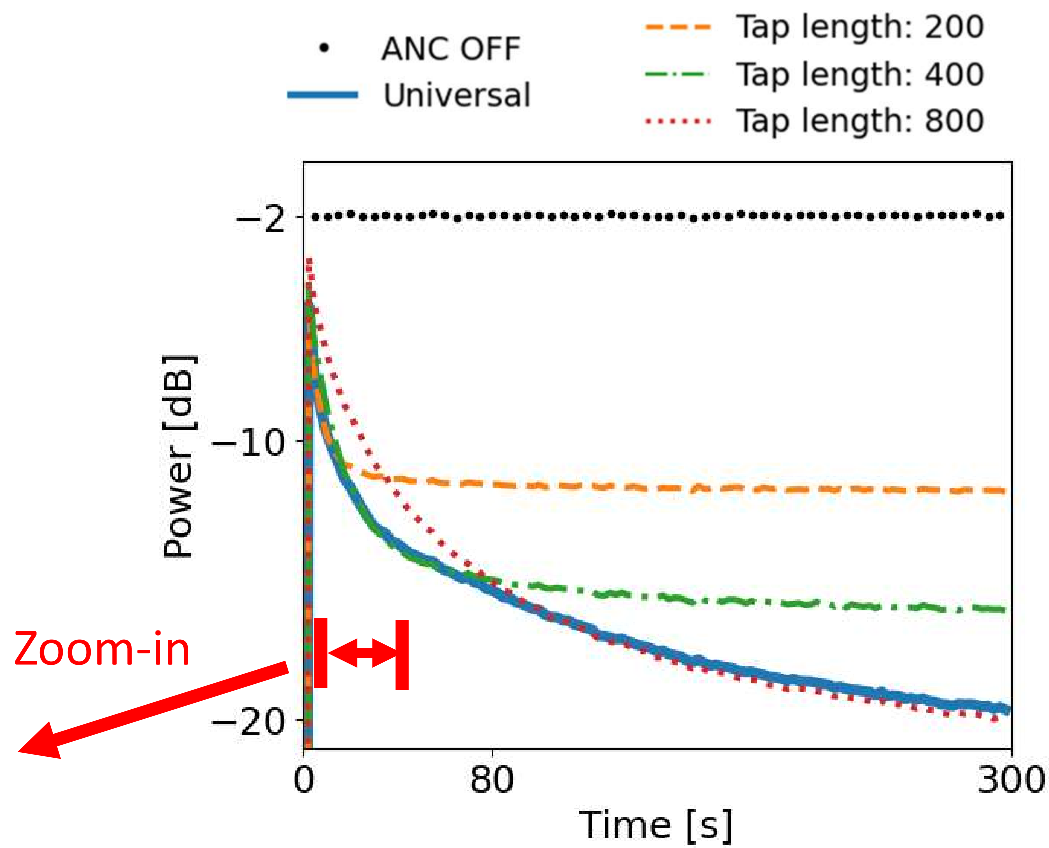


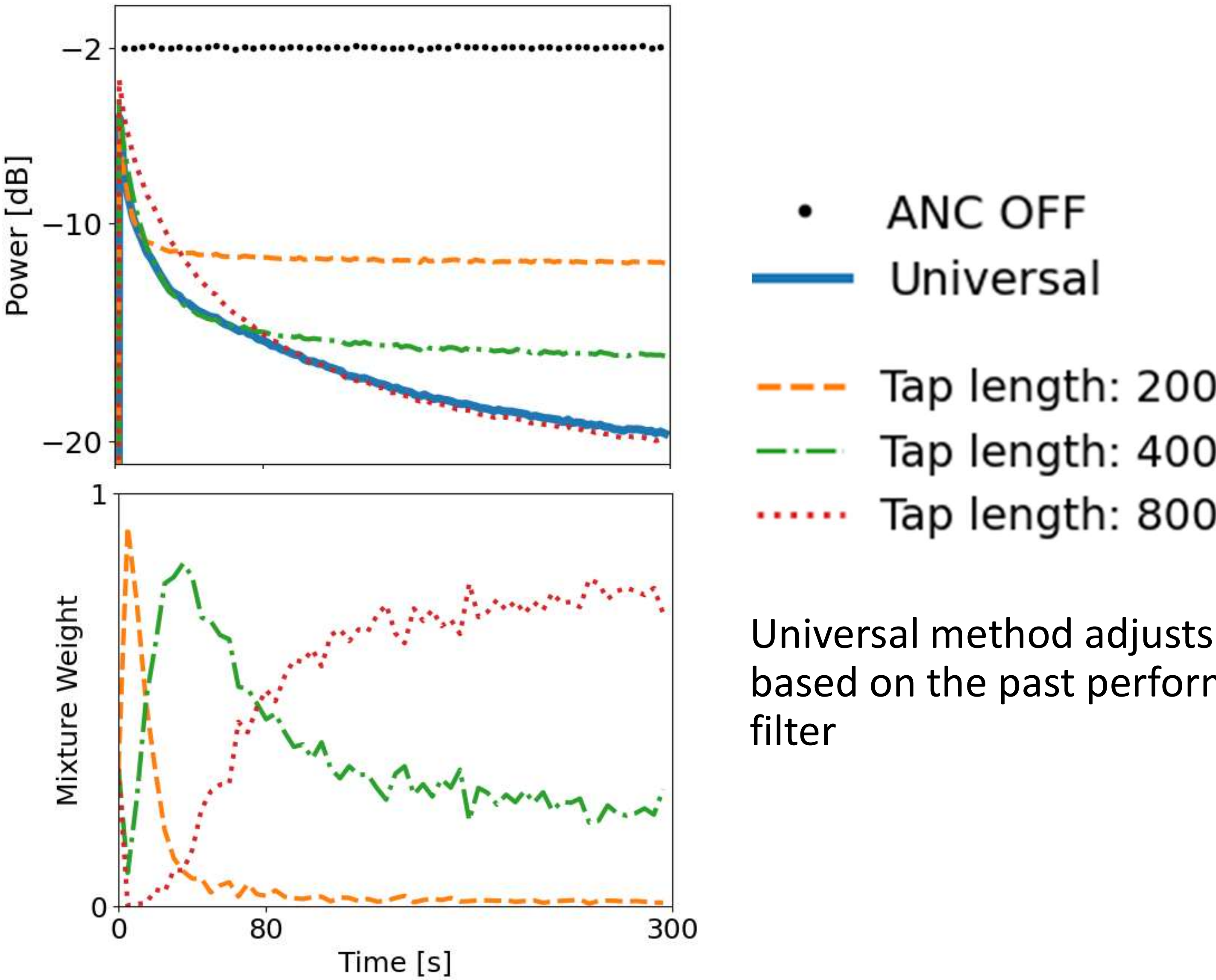
Control source → Error mic

Universal method can still track the best candidate filter in both transient and steady-state phase

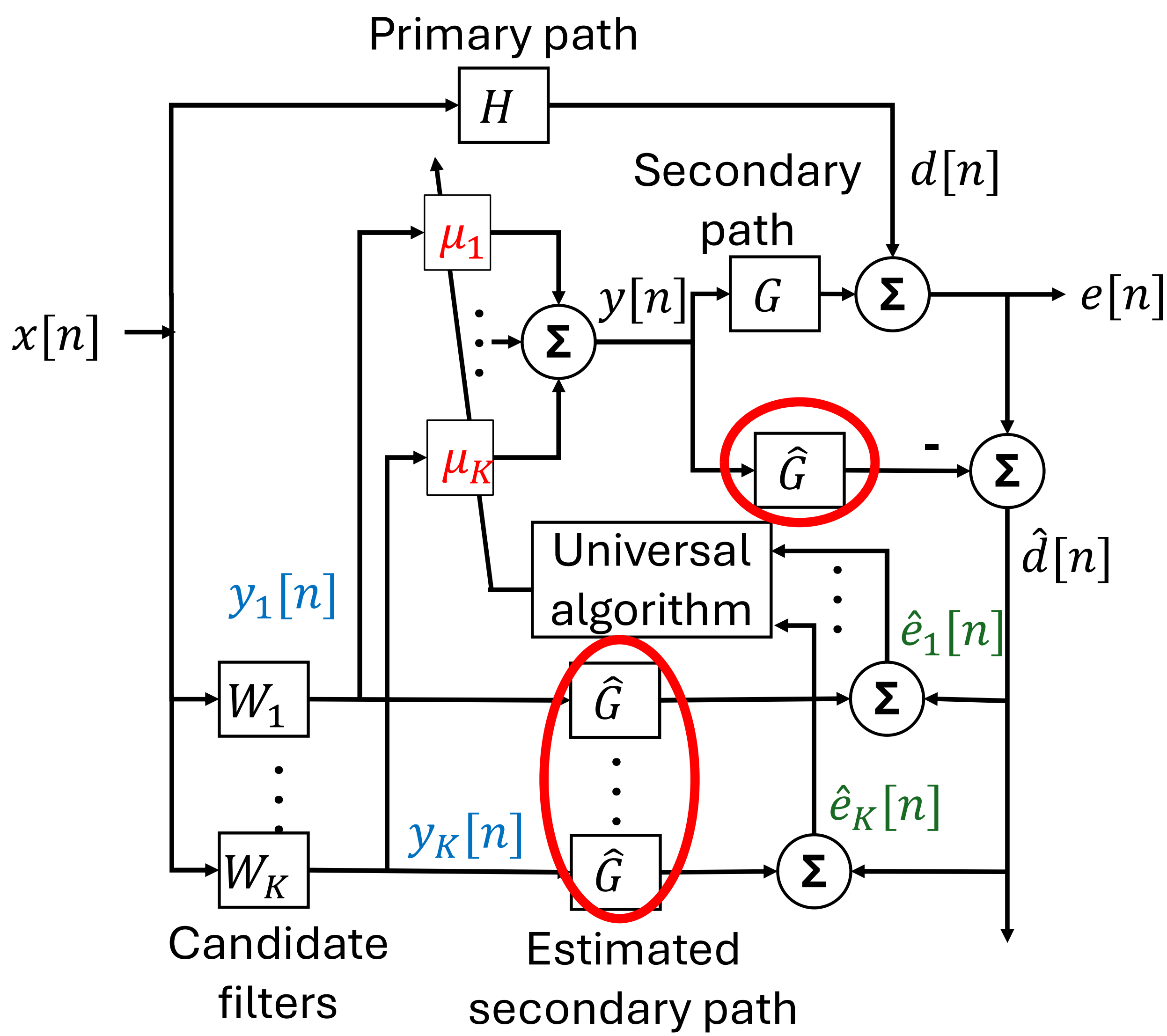
100 Monte Carlo trials:







Universal method adjusts the mixture weights based on the past performance of each candidate filter



- What if the estimated secondary paths are not perfectly accurate?

  A noisy secondary path:

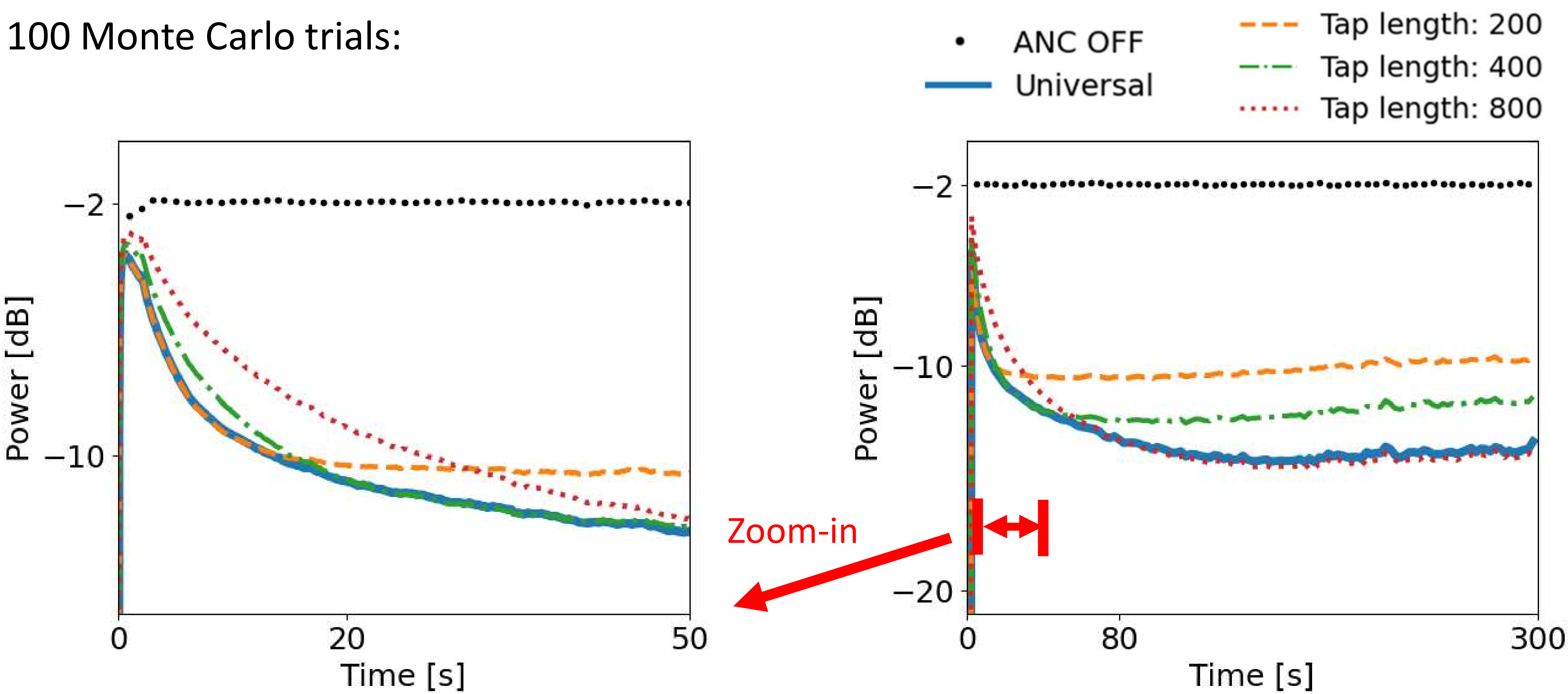
$$\hat{G} = G + \nu$$

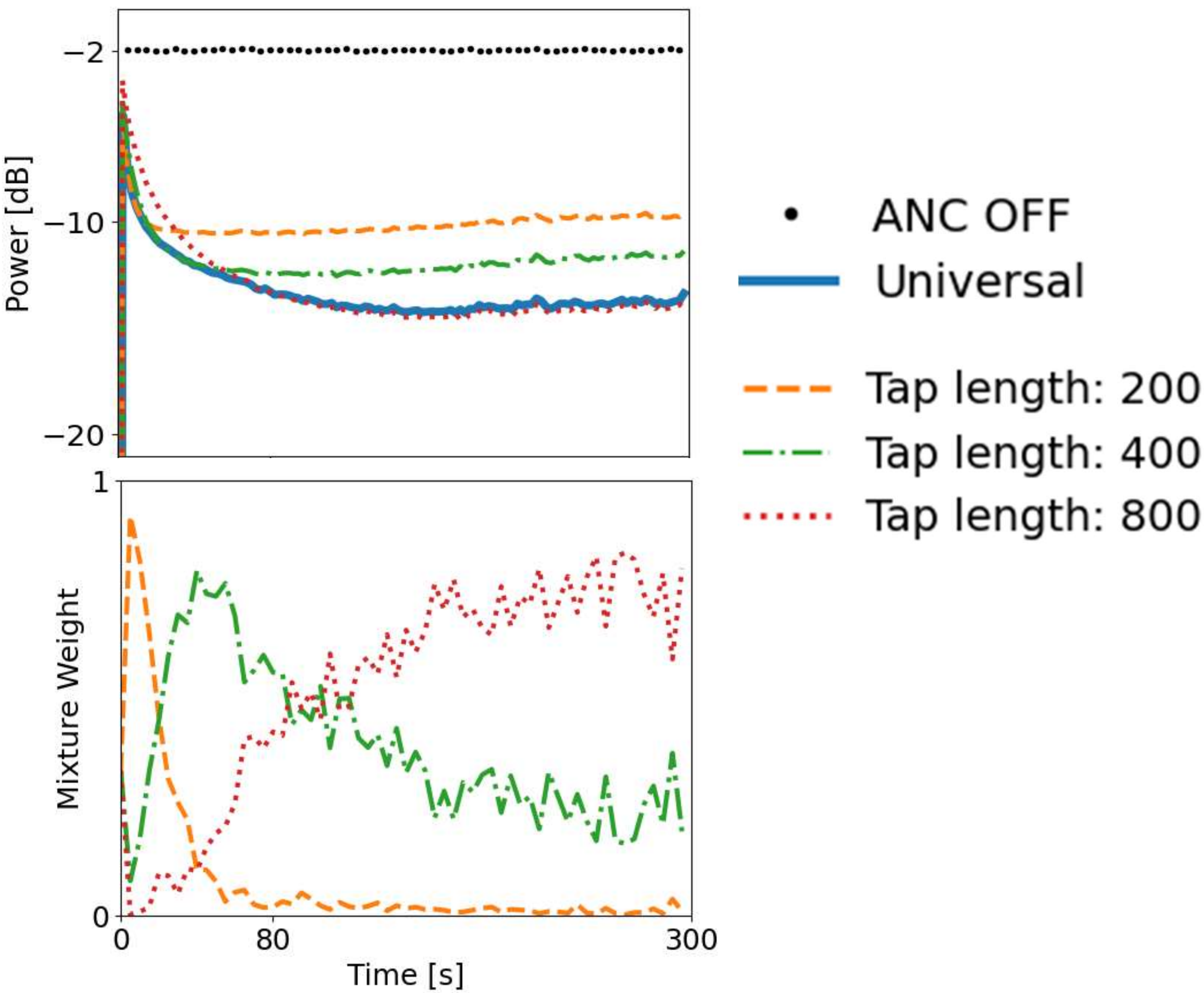
We can check the performance in a high perturbance:

Pow(G) / Pow(ν) = 3.3

Universal method can still track the best candidate filter in a perturbed secondary path case

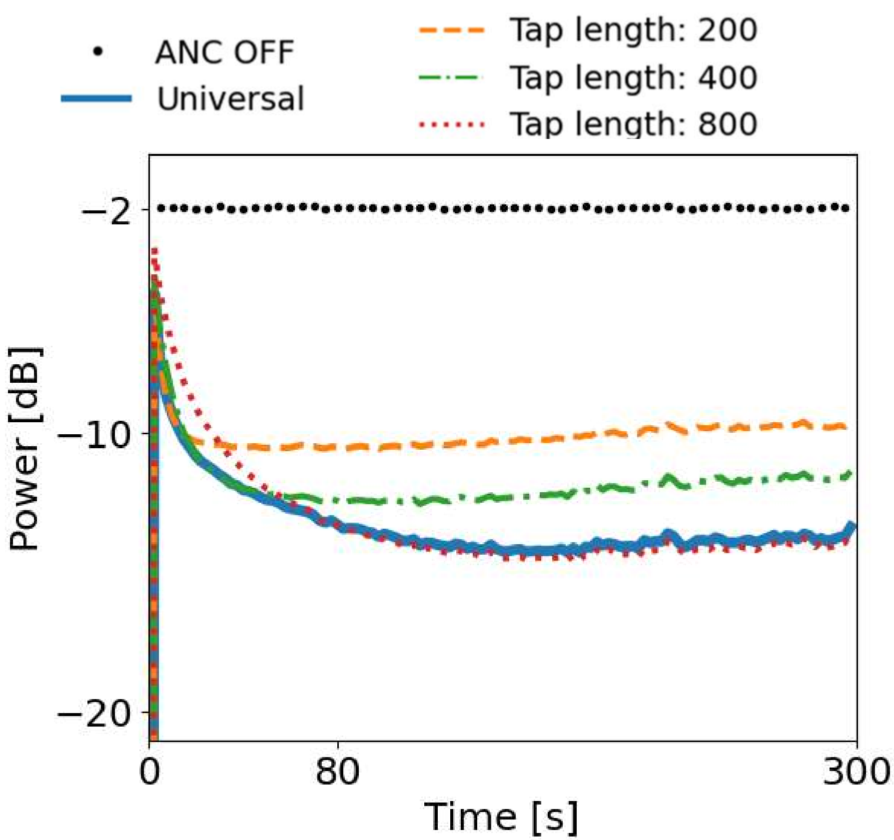
100 Monte Carlo trials:





−2
−10
−20
Power [dB]
1
0
Mixture Weight
0
80
300
Time [s]
ANC OFF
Universal
Tap length: 200
Tap length: 400
Tap length: 800

Conclusion



- Blending the outputs of candidate filters based on their past performance
- Good performance under secondary path perturbation
- In the future:
  - Other filters
  - Efficient implementation
  - Real-time experiments