UNIVERSITÀ DEGLI STUDI DI BERGAMO
Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Reproduction of Results from the Paper:

"A Robust Variable Forgetting Factor Recursive Least-Squares Algorithm for System Identification"

Academic Year: 2024 - 25

Master Degree in COMPUTER ENGINEERING

Data Science and Data Engineering Curriculum

Course:
Adaptive Learning, Estimation And Supervision Of Dynamical Systems

Professor
Prof. Mirko Mazzoleni

Student
Arash Abedi

Introduction and Problem Statement

I. What is the problem?

- System identification in the presence of noise.
- The goal is to estimate the impulse response of an unknown system using an adaptive filter.
- Challenges:
  - Noise corrupts the output of the unknown system.
  - Trade-off between tracking capabilities and stability in adaptive filtering.

II. Why is it important?

- Applications in echo cancellation, noise reduction, and channel estimation.
- Adaptive filters are widely used in real-time signal processing.

Overview of the Paper

- **Authors :**
  - Constantin Paleologu, Jacob Benesty, Silviu Ciochină.
- **Key Contribution :**
  - Proposes a Variable Forgetting Factor Recursive Least-Squares (VFF-RLS) algorithm for system identification.
  - Improves tracking capabilities while maintaining stability and low misadjustment.
- **Main Idea :**
  - The forgetting factor (λ) is adjusted dynamically based on the system noise and error signal.
  - Ensures fast convergence and robustness to noise.

Problem Setting

- **System Model :**
  - Unknown system: FIR filter with impulse response h.
  - Input signal: $x(n)$ (white Gaussian noise or AR(1) process)
  - AR(1) is generated by filtering white noise through a first-order autoregressive model.
  - Output signal: $y(n) = h^T x(n) + v(n)$, where $v(n)$ is additive noise.
- **Objective :**
  - Estimate h using an adaptive filter w(n).
  - Minimize the misalignment between h and w(n).
- **Challenges :**
  - Noise $v(n)$ corrupts the output.
  - Trade-off between tracking speed and stability.

Classical RLS Algorithm: Key Equations

- **Error Signal:**

$$\mathbf{e}(\boldsymbol{n}) = \mathbf{d}(\boldsymbol{n}) - \boldsymbol{w}^{\boldsymbol{T}}(\boldsymbol{n}-\mathbf{1})\boldsymbol{x}(\boldsymbol{n})$$

  - where:

    $\mathrm{d}(n)$: Desired signal (output of the unknown system + noise).

    w(n−1): Filter coefficients at time n−1.

    $x(n)$: Input signal vector at time n.

- **Kalman Gain:**

$$\boldsymbol{k}(\boldsymbol{n}) = \frac{\boldsymbol{P}(\boldsymbol{n}-\mathbf{1})\boldsymbol{x}(\boldsymbol{n})}{\boldsymbol{\lambda}+\boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{n})\boldsymbol{P}(\boldsymbol{n}-\mathbf{1})\boldsymbol{x}(\boldsymbol{n})}$$

  - where:

    $P(n-1)$: Inverse correlation matrix at time n−1.

    λ: Forgetting factor (constant in classical RLS).

    $x(n)$: Input signal vector at time n.

Classical RLS Algorithm: Key Equations

- **Filter Update:**

$$\boldsymbol{w}(\boldsymbol{n}) = \boldsymbol{w}(\boldsymbol{n}-\boldsymbol{1}) + \boldsymbol{k}(\boldsymbol{n})\boldsymbol{e}(\boldsymbol{n})$$

  - where:

    $w(n-1)$: Filter coefficients at time n−1.

    $k(n)$: Kalman gain at time n.

    $e(n)$: Error signal at time n.

- **Inverse Correlation Matrix Update:**

$$\boldsymbol{P}(\boldsymbol{n}) = \frac{\boldsymbol{1}}{\boldsymbol{\lambda}}\left(\boldsymbol{P}(\boldsymbol{n}-\boldsymbol{1}) - \boldsymbol{k}(\boldsymbol{n})\boldsymbol{x}^{\boldsymbol{T}}(\boldsymbol{n})\boldsymbol{P}(\boldsymbol{n}-\boldsymbol{1})\right)$$

  - where:

    $P(n-1)$: Inverse correlation matrix at time n−1.

    $k(n)$: Kalman gain at time n.

    $x(n)$: Input signal vector at time n.

    λ: Forgetting factor (constant in classical RLS).

VFF-RLS Algorithm: Key Equations

- **Forgetting Factor Update:**

$$\lambda(n) = MIN\left(\frac{\sigma_q(n)\sigma_v}{\sigma_e(n) - \sigma_v}, \lambda_{max}\right)$$

  - where:

    $\sigma_q(n)$: Power estimate of the intermediate variable q(n).

    σ_v : Power estimate of the system noise v(n).

    σ_e (n): Power estimate of the error signal e(n).

    $\lambda_{\max}$: Maximum value of the forgetting factor (close to 1).

- **Error Signal Power:**

$$\sigma_e^2(n) = \alpha\sigma_e^2(n-1) + (1-\alpha)e^2(n)$$

  - where:

    α: Weighting factor for the exponential window ($\alpha = 1 - 1/k_\alpha M$).

    $e(n)$: Error signal at time n.

VFF-RLS Algorithm: Key Equations

- **Intermediate Variable Power:** $$\boldsymbol{\sigma_q^2(n) = \alpha\sigma_q^2(n-1) + (1-\alpha)q^2(n)}$$
  - where:

    α: Weighting factor for the exponential window.

    $q(n)$: Intermediate variable, $q(n) = x_{(n)}^{T} p(n-1)x(n)$

- **System Noise Power:** $$\boldsymbol{\sigma_v^2(n) = \beta\sigma_v^2(n-1) + (1-\beta)e^2(n)}$$
  - where:

    β: Weighting factor for the exponential window (β $= 1 - 1/k_{\beta}M$).

    $e(n)$: Error signal at time n.

Proposed Solution: VFF-RLS Algorithm

- **Classical RLS Limitations :**
  - Fixed forgetting factor (λ) leads to a trade-off between tracking and stability.
- **VFF-RLS Algorithm :**
  - Dynamic Forgetting Factor :
    - $\lambda\,(n)$ is adjusted based on the system noise and error signal.
    - Ensures fast tracking during changes and low misadjustment in steady-state.
  - Key Equations :
    - Forgetting factor update: $\lambda(n) = MIN\left(\frac{\sigma_q(n)\sigma_v}{\sigma_e(n) - \sigma_v}, \lambda_{max}\right)$
    - Power estimates: $\sigma_e^2, \sigma_q^2, \sigma_v^2$
  - Advantages :
    - Robust to noise and system changes, simple and computationally efficient.

Implementation Details

- **Steps :**

  1. Generate input signal $\mathrm{X}(n)$ (white Gaussian noise or AR(1) process).
  2. AR(1) Process: $x(n) = 0.9x(n-1) + w(n)$
  3. Simulate the unknown system using a FIR filter h.
  4. Add noise to the output to create the desired signal $\mathrm{d}(n)$.
  5. Implement RLS and VFF-RLS algorithms to estimate h.

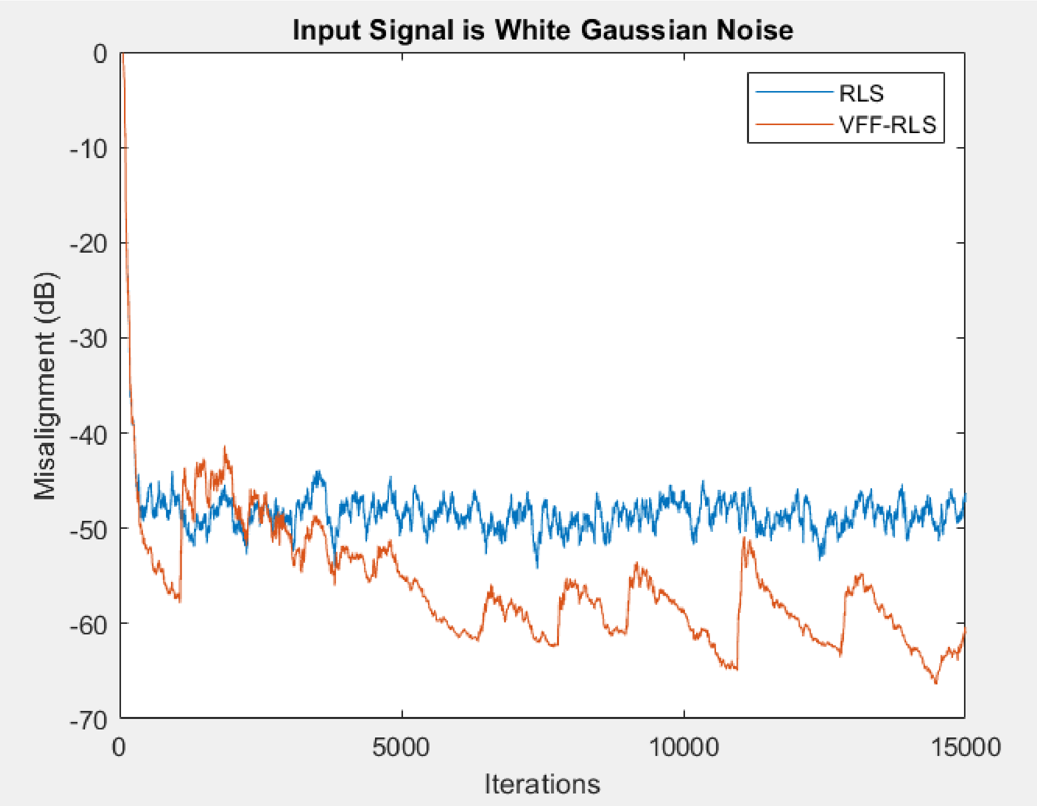
- **Key Parameters :**

  - Filter length: M = 64
  - Forgetting factor: $\lambda = 1 - \frac{1}{3M}$ (for RLS)
  - SNR: 20 dB

Simulation Results (White Gaussian Noise Input)

A. Misalignment Comparison (RLS vs VFF-RLS)

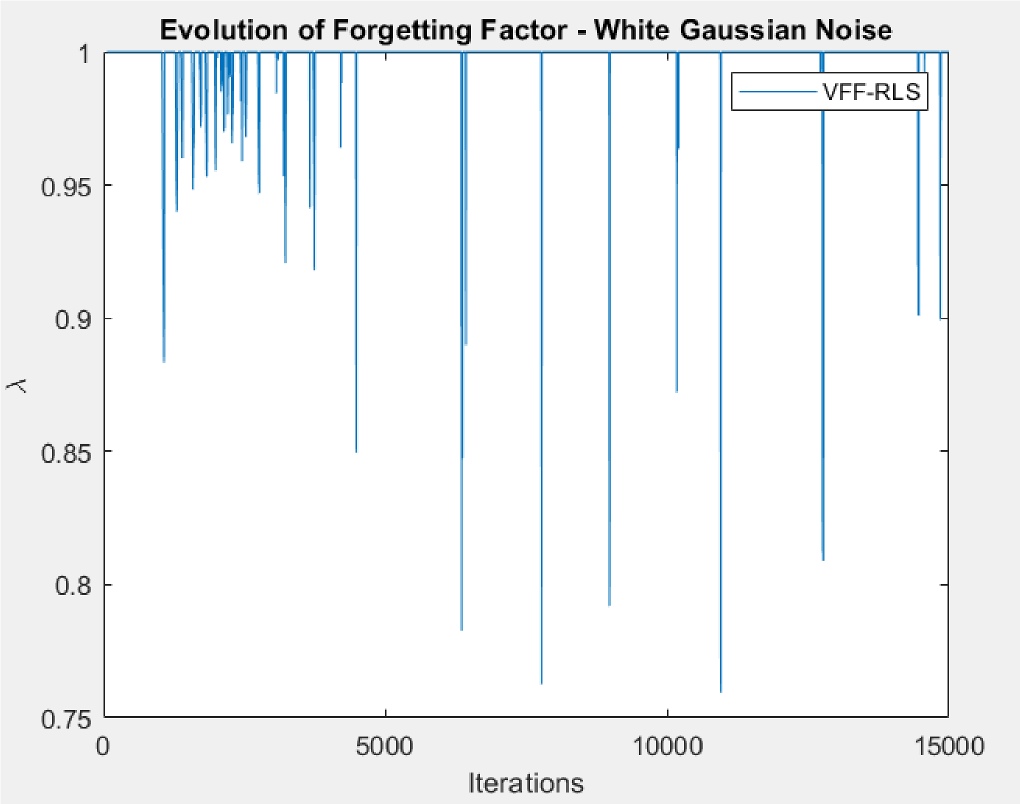
- The misalignment (in dB) is plotted over iterations for both RLS and VFF-RLS algorithms.
- VFF-RLS achieves lower misalignment compared to RLS, demonstrating better performance in system identification.
- VFF-RLS shows faster convergence and better tracking capabilities.
- Highlights the superiority of VFF-RLS in handling noise and maintaining low misadjustment.



Simulation Results (White Gaussian Noise Input)

B. Evolution of Forgetting Factor (λ)

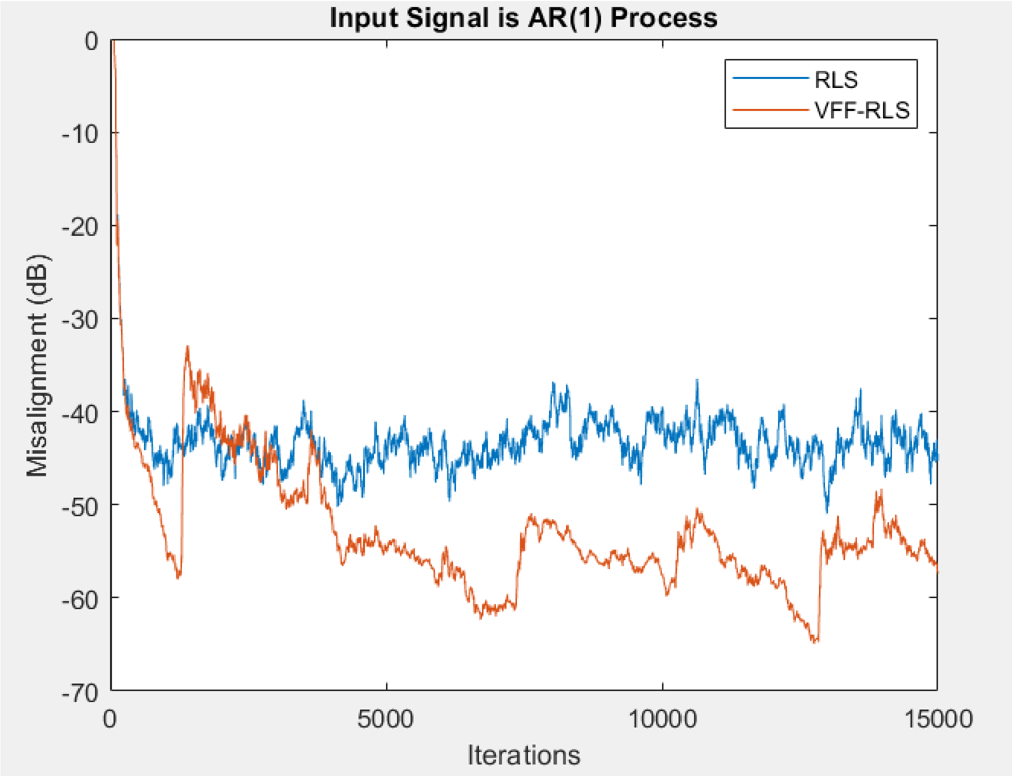
- The forgetting factor λ(n) is plotted over iterations for the VFF-RLS algorithm.
- λ(n) adapts dynamically based on the system noise and error signal.
- It decreases during abrupt changes (if any) and stabilizes in steady-state.
- Demonstrates the adaptive nature of VFF-RLS, which improves tracking and stability.



Simulation Results (AR(1) Process Input)

A. Misalignment Comparison (RLS vs VFF-RLS)

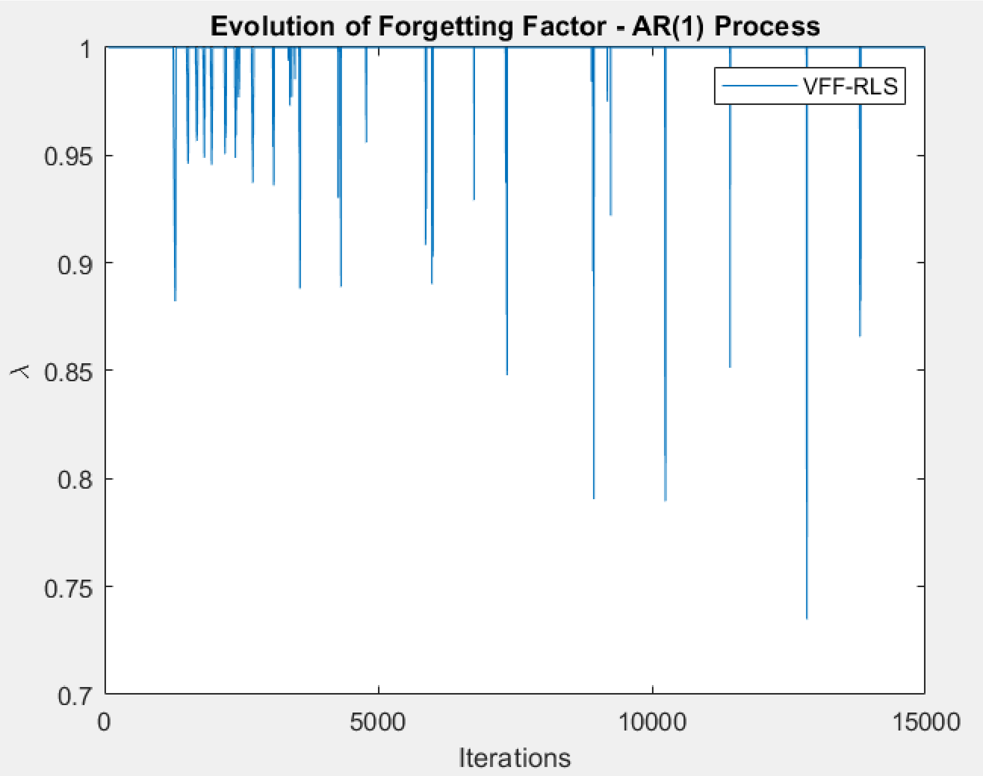
- The misalignment (in dB) is plotted over iterations for both RLS and VFF-RLS algorithms.
- VFF-RLS outperforms RLS in terms of misalignment, especially during abrupt changes.
- The adaptive nature of VFF-RLS allows it to handle correlated input signals AR(1) more effectively.
- Shows that VFF-RLS is robust even with correlated input signals.



Simulation Results (AR(1) Process Input)

B. Evolution of Forgetting Factor (λ)

- The forgetting factor λ(n) is plotted over iterations for the VFF-RLS algorithm.
- λ(n) adapts dynamically, showing faster convergence during changes and stability in steady-state.
- Demonstrates the effectiveness of the variable forgetting factor in handling correlated input signals.



Discussion and Insights

- **Why Does VFF-RLS Work Better?**
  - Dynamic $\lambda(n)$ ensures fast tracking during changes and low misadjustment in steady-state.
  - Robust to noise and system variations.
- **Limitations :**
  - Requires accurate estimation of noise power.
  - Slightly higher computational complexity than RLS.
- **Future Work :**
  - Apply to real-world problems like echo cancellation or channel estimation.

Conclusion

- **Summary**
  - Reproduced the results of the paper successfully.
  - VFF-RLS provides a robust solution for system identification in noisy environments.
- **Key Takeaways :**
  - Dynamic forgetting factor improves performance.
  - Simple and effective algorithm for real-time applications.