Machine Learning Algorithms

LMS Analysis

Markus Rupp

14.8.2020

LMS Algorithm

- Let's consider the steepest descent iteration:

$$\underline{\hat{w}}_k = \underline{\hat{w}}_{k-1} + \mu\left(\underline{r}^*_{\underline{\mathbf{x}}\mathbf{d}} - R^*_{\underline{\underline{\mathbf{xx}}}}\,\underline{\hat{w}}_{k-1}\right);\ k = 1,2,\ldots$$

- Obviously, the knowledge of the cross-correlation vector and the ACF matrix is required.
- If this knowledge is not present, we could operate with estimates. The most simple ones are:

$$\hat{R}_{\underline{\underline{\mathbf{xx}}}} = \underline{\mathbf{x}}_k\,\underline{\mathbf{x}}_k^H; \qquad \underline{\hat{r}}_{\underline{\mathbf{x}}\mathbf{d}} = \underline{\mathbf{x}}_k\mathbf{d}_k^*$$

$$\underline{\mathbf{x}}_k^T = \left[\mathbf{x}(k), \mathbf{x}(k-1), \ldots, \mathbf{x}(k-M+1)\right]$$

LMS Algorithm

- They are called instantaneous estimates.
- Assuming stationary processes, also other estimates are possible, for example

$$\hat{R}_{\underline{\mathbf{xx}}} = \frac{1}{N}\sum_{l=0}^{N-1} \underline{\mathbf{x}}_{k-l}\,\underline{\mathbf{x}}_{k-l}^{H}; \qquad \hat{\underline{r}}_{\underline{\mathbf{x}}\mathbf{d}} = \frac{1}{N}\sum_{l=0}^{N-1} \underline{\mathbf{x}}_{k-l}\mathbf{d}_{k\text{-}l}^{*}$$

- We will see in the next chapter that this choice leads to a special form, the so-called RLS algorithm.
- Plugging in the steepest descent iterations the instantaneous estimates, we obtain:

$$\underline{\hat{\mathbf{w}}}_k = \underline{\hat{\mathbf{w}}}_{k-1} + \mu\underline{\mathbf{x}}_k^{*}\left(\mathbf{d}_k - \underline{\mathbf{x}}_k^{T}\underline{\hat{\mathbf{w}}}_{k-1}\right); k = 1,2,\ldots$$

LMS Algorithm

- Utilizing such instantaneous values,

$$\underline{\hat{\mathbf{w}}}_k = \underline{\hat{\mathbf{w}}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \underbrace{\left(\mathbf{d}_k - \underline{\mathbf{x}}_k^T \underline{\hat{\mathbf{w}}}_{k-1}\right)}_{\tilde{\mathbf{e}}_{a,k}}; \; k = 1,2,\ldots$$

- The update equation of the LMS algorithm is obtained.
- The error signal

$$\tilde{\mathbf{e}}_{a,k} = \mathbf{d}_k - \underline{\mathbf{x}}_k^T \underline{\hat{\mathbf{w}}}_{k-1}$$

- is called the disturbed/distorted a-priori error signal.
- Due to the reference model, there is also an undisturbed a-priori error signal:

$$\mathbf{e}_{a,k} = \underline{\mathbf{x}}_k^T \underline{\mathbf{w}}_o - \underline{\mathbf{x}}_k^T \underline{\hat{\mathbf{w}}}_{k-1} = \underline{\mathbf{x}}_k^T \underline{\tilde{\mathbf{w}}}_{k-1}$$

LMS Algorithm

- Correspondingly, would we use the estimates at time instant k rather than k-1, we call the error a-posteriori error signals:

$$\tilde{\mathbf{e}}_{p,k} = \mathbf{d}_k - \underline{\mathbf{x}}_k^T \underline{\hat{\mathbf{w}}}_k$$

$$\mathbf{e}_{p,k} = \underline{\mathbf{x}}_k^T \underline{\mathbf{w}}_o - \underline{\mathbf{x}}_k^T \underline{\hat{\mathbf{w}}}_k = \underline{\mathbf{x}}_k^T \underline{\tilde{\mathbf{w}}}_k$$

- Note that due to the nature of the recursive algorithm, the estimates are random processes:

$$\underline{\hat{\mathbf{w}}}_k = \underline{\hat{\mathbf{w}}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \left(\mathbf{d}_k - \underline{\mathbf{x}}_k^T \underline{\hat{\mathbf{w}}}_{k-1}\right); k = 1,2,\ldots$$

LMS Algorithm

- Many different choices for the step-size are possible:

$$\textbf{arbitrary } \mu_k : \quad \textbf{stochastic gradient algorithm}$$

$$\mu_k = \frac{\alpha}{\|\underline{\mathbf{x}}_k\|_2^2} \quad \textbf{with } \alpha > 0, \textbf{(data-) Normalized LMS algorithm} = \textbf{NLMS}$$

$$\mu_k = \frac{\alpha}{1+\alpha\|\underline{\mathbf{x}}_k\|_2^2} \quad \textbf{with } \alpha > 0, \textbf{a-posteriori form of the LMS algorithm}$$

$$\mu_k = \frac{\alpha}{\varepsilon+\|\underline{\mathbf{x}}_k\|_2^2} \quad \textbf{with } \varepsilon > \mathbf{0}, \textbf{the so-called } \varepsilon-\textbf{NLMS algorithm}$$

LMS Algorithm

- Its popularity is also deserved by its mathematical simplicity. Not only

$$\mathrm{E}\left[\left|\tilde{\mathrm{e}}_{a,k}\right|^2\right]$$

- Can be minimized but also

$$\mathrm{E}\left[\left|f\left[\tilde{\mathrm{e}}_{a,k}\right]\right|^2\right]$$

- With arbitrary nonlinear functions f[]. For example $f[x]=x^{K/2}$ leads to the well known Least-Mean-K algorithm (LMK):

$$\underline{\hat{\mathbf{w}}}_k = \underline{\hat{\mathbf{w}}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \left|\widetilde{\mathbf{e}}_{a,k}\right|^{\mathrm{K}\text{-}2} \widetilde{\mathbf{e}}_{a,k}$$

LMS Algorithm

- Other variants of the LMS algorithm are:

$$\textbf{Least}-\textbf{Mean Fourth} \quad : \underline{\hat{\mathbf{w}}}_k = \underline{\hat{\mathbf{w}}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \left| \tilde{\mathbf{e}}_{a,k} \right|^2 \tilde{\mathbf{e}}_{a,k}$$

$$\textbf{Least}-\textbf{Mean Mixed Norm} : \underline{\hat{\mathbf{w}}}_k = \underline{\hat{\mathbf{w}}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \left(\beta + (1-\beta) \left| \tilde{\mathbf{e}}_{a,k} \right|^2 \right) \tilde{\mathbf{e}}_{a,k}$$

$$\textbf{Leaky-LMS :} \quad \underline{\hat{\mathbf{w}}}_k = \beta \underline{\hat{\mathbf{w}}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \tilde{\mathbf{e}}_{a,k}$$

$$\textbf{Sign-error LMS :} \quad \underline{\hat{\mathbf{w}}}_k = \underline{\hat{\mathbf{w}}}_{k-1} + \mu \underline{\mathbf{x}}_k^* \,\mathbf{sgn}\left[\tilde{\mathbf{e}}_{a,k} \right]$$

LMS Algorithm

- Since the LMS algorithm uses instantaneous estimates of cross correlation vector and ACF matrix, we cannot expect that the algorithm behaves identical to the steepest descent algorithm.
- We therefore have to analyze the algorithm by computing its behavior in mean and mean square.

LMS Algorithm Analysis

- Independence assumptions
  - The observation of d_k originates from a reference model: $\mathbf{d}_k = \underline{w}_o^T \underline{\mathbf{x}}_k + \mathbf{v}_k$, $\mathbf{v}_k$ and $\underline{\mathbf{x}}_k$ are of zero mean.
  - The regression vectors $\underline{\mathbf{x}}_k$ are statistically independent for different time instants: $f_{\underline{\mathbf{xx}}}(\underline{\mathbf{x}}_k,\underline{\mathbf{x}}_l)=f_{\underline{\mathbf{xx}}}(\underline{\mathbf{x}}_k)\, f_{\underline{\mathbf{xx}}}(\underline{\mathbf{x}}_l)$ for k and l different.
  - The input process $\mathbf{x}_k$ is spherically invariant and Gaussian distributed with zero mean.
  - The additive noise $\mathbf{v}_k$ is statistically independent of the driving process $\mathbf{x}_k$.
- Note that through these conditions, $\underline{\mathbf{w}}_k$ is statistically independent of $\underline{\mathbf{x}}_l$;l>k.

Spherically Invariant Processes

- A complex-valued variable **z=x+jy** is called Gaussian when **x** and **y** are joint Gaussian.
- The second moment of **z** is given by:

$$R_{\mathbf{zz}} = \mathrm{E}\left[\mathbf{zz}^*\right] = R_{\mathbf{xx}} + R_{\mathbf{yy}} + j\left(R_{\mathbf{yx}} - R_{\mathbf{xy}}\right)$$

- Note that the information of $R_{\mathbf{zz}}$ is not sufficient to conclude back to $R_{\mathbf{xx}}$,$R_{\mathbf{yy}}$ and $R_{\mathbf{xy}}$. For this more information is required, for example:

$$R_{\mathbf{zz}^*} = \mathrm{E}\left[\mathbf{zz}\right] = R_{\mathbf{xx}} - R_{\mathbf{yy}} + j\left(R_{\mathbf{yx}} + R_{\mathbf{xy}}\right)$$

- For spherically invariant processes, we have $R_{\mathbf{zz*}}$=0, or equivalently $R_{\mathbf{xx}}$=$R_{\mathbf{yy}}$ and $R_{\mathbf{xy}}$=- $R_{\mathbf{yx}}$.
- The joint density function of a vector $\underline{z}$ is thus given by:

$$f_{\underline{\mathbf{x,y}}}(\underline{x}, \underline{y}) = f_{\underline{\mathbf{z}}}(\underline{z}) = \frac{1}{\pi^p} \frac{1}{\det(R_{\underline{\mathbf{zz}}})} \exp\left(- \underline{z}^H R_{\underline{\mathbf{zz}}}^{-1} \underline{z}\right)$$

LMS Algorithm Analysis

- The parameter error vector in the mean:

$$\underline{\tilde{\mathbf{w}}}_k = \underline{w}_o - \underline{\hat{\mathbf{w}}}_k$$

$$\underline{\tilde{\mathbf{w}}}_k = \left(I - \mu \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right)\underline{\tilde{\mathbf{w}}}_{k-1} - \mu \underline{\mathbf{x}}_k^* \mathbf{v}_k$$

$$\mathrm{E}\left[\underline{\tilde{\mathbf{w}}}_k\right] = \mathrm{E}\left[\left(I - \mu \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right)\underline{\tilde{\mathbf{w}}}_{k-1}\right] - \mu\, \mathrm{E}\left[\underline{\mathbf{x}}_k^* \mathbf{v}_k\right]$$

- With the independence assumptions:

$$\mathrm{E}\left[\underline{\tilde{\mathbf{w}}}_k\right] = \left(I - \mu R_{\underline{\mathbf{xx}}}^*\right)\mathrm{E}\left[\underline{\tilde{\mathbf{w}}}_{k-1}\right]$$

- Behavior identical to steepest descent!

LMS Algorithm Analysis

- Behavior identical to steepest descent!

$$\mathrm{E}\left[\underline{\tilde{\mathbf{w}}}_k\right] = \left(I - \mu R^*_{\underline{\mathbf{xx}}}\right)\mathrm{E}\left[\underline{\tilde{\mathbf{w}}}_{k-1}\right]$$

- Thus convergence guaranteed for

$$0 < \mu < \frac{2}{\lambda_{\max}}$$

LMS Algorithm Analysis

- The parameter error vector in the mean square sense:

$$
\begin{aligned}
P_k &= \mathrm{E}\left[\left(\underline{w}_o - \underline{\hat{\mathbf{w}}}_k\right)\left(\underline{w}_o - \underline{\hat{\mathbf{w}}}_k\right)^H\right] = \mathrm{E}\left[\underline{\tilde{\mathbf{w}}}_k \underline{\tilde{\mathbf{w}}}_k^H\right] \\
&= \mathrm{E}\left[\left(I - \mu \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right) P_{k-1} \left(I - \mu \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right)^H\right] + \mu^2\, \mathrm{E}\left[\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T \left|\mathbf{v}_k\right|^2\right] \\
&= P_{k-1} - \mu\, \mathrm{E}\left[\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T P_{k-1}\right] - \mu\, \mathrm{E}\left[P_{k-1} \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right] + \mu^2\, \mathrm{E}\left[\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T P_{k-1} \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right] \\
&\quad + \mu^2\, \mathrm{E}\left[\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T \left|\mathbf{v}_k\right|^2\right]
\end{aligned}
$$

Univ.-Prof. Dr.-Ing. Markus Rupp

LMS Algorithm Analysis

- Most problematic is the term

$$\mathrm{E}\left[\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T P_{k-1} \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right]$$

- It can be solved explicitly for (spherically) Gaussian processes. We distinguish real-valued processes:

$$\mathrm{E}\left[\underline{\mathbf{x}}_k \underline{\mathbf{x}}_k^T P_{k-1} \underline{\mathbf{x}}_k \underline{\mathbf{x}}_k^T\right] = 2R_{\underline{\underline{\mathbf{xx}}}} P_{k-1} R_{\underline{\underline{\mathbf{xx}}}} + R_{\underline{\underline{\mathbf{xx}}}} \operatorname{trace}\left(P_{k-1} R_{\underline{\underline{\mathbf{xx}}}}\right)$$

- Complex-valued processes:

$$\mathrm{E}\left[\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T P_{k-1} \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right] = R_{\underline{\underline{\mathbf{xx}}}}^* P_{k-1} R_{\underline{\underline{\mathbf{xx}}}}^* + R_{\underline{\underline{\mathbf{xx}}}}^* \operatorname{trace}\left(P_{k-1} R_{\underline{\underline{\mathbf{xx}}}}^*\right)$$

LMS Algorithm Analysis

- The procedure can be extended to general spherically invariant processes. In the general case we obtain

$$\mathrm{E}\left[\underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T P_{k-1} \underline{\mathbf{x}}_k^* \underline{\mathbf{x}}_k^T\right] = \gamma R_{\underline{\mathbf{xx}}}^* P_{k-1} R_{\underline{\mathbf{xx}}}^* + R_{\underline{\mathbf{xx}}}^* \operatorname{trace}\left(P_{k-1} R_{\underline{\mathbf{xx}}}^*\right)$$

- with some positive constant γ=1 for complex and γ=2 for real valued processes.   .
- The update recursion for the parameter error covariance matrix reads now:

$$\begin{aligned} P_k = P_{k-1} &- \mu R_{\underline{\mathbf{xx}}}^* P_{k-1} - \mu P_{k-1} R_{\underline{\mathbf{xx}}}^* + \mu^2 \gamma R_{\underline{\mathbf{xx}}}^* P_{k-1} R_{\underline{\mathbf{xx}}}^* \\ &+ \mu^2 R_{\underline{\mathbf{xx}}}^* \operatorname{trace}\left(P_{k-1} R_{\underline{\mathbf{xx}}}^*\right) + \mu^2 R_{\underline{\mathbf{xx}}}^* \sigma_{\mathbf{v}}^2 \end{aligned}$$

LMS Algorithm Analysis

- Orthogonalization for the ACF matrix leads to:

$$Q^H R_{\underline{\mathbf{xx}}}^* Q = \Lambda; \qquad Q^H P_k Q = C_k$$

- We thus obtain:

$$\begin{aligned} C_k &= C_{k-1} - \mu\Lambda C_{k-1} - \mu C_{k-1}\Lambda + \mu^2\gamma\Lambda C_{k-1}\Lambda \\ &\quad + \mu^2\Lambda\,\mathrm{trace}\left[\Lambda C_{k-1}\right] + \mu^2\Lambda\sigma_{\mathbf{v}}^2 \end{aligned}$$

- But note that:

$$\mathrm{E}\left[\left\|\underline{\tilde{\mathbf{w}}}_k\right\|_2^2\right] = \mathrm{trace}\left[P_k\right] = \mathrm{trace}\left[C_k\right]$$

LMS Algorithm Analysis

- Thus, only the diagonal terms of C_k are of importance. Order diagonal terms in vector $\underline{c}_k$:

$$\underline{c}_k = B\underline{c}_{k-1} + \mu^2 \sigma_{\mathbf{v}}^2 \underline{\lambda}$$

$$B = I - 2\mu\Lambda + \mu^2 \gamma \Lambda^2 + \mu^2 \underline{\lambda}\underline{\lambda}^T$$

$$= \begin{cases} 1 - 2\mu\lambda_i + \mu^2 (1+\gamma)\lambda_i^2 & \textbf{main diagonal} \\ \mu^2 \lambda_i \lambda_j & \textbf{else} \end{cases}$$

LMS Algorithm Analysis

- **Theorem 3.1:** Under the given conditions, the LMS algorithm is convergent in the mean-square sense, when:

$$0 < \mu < \frac{2}{\gamma\lambda_{\max} + \mathrm{trace}(\Lambda)}$$

- **Proof:** Convergence is guaranteed if the eigenvalues of matrix B are upper bounded by one. Since B is positive definite and symmetrical, all its eigenvalues are real-valued and positive.

LMS Algorithm Analysis

- **Proof:** A sufficient condition is that the largest eigenvalue is smaller than one. This is the l_2 norm of the matrix B. An even loser condition is that the l_1 norm of B is smaller than one:

$$\lambda_{\max}^{(B)} = \left\| B \right\|_2 \leq \left\| B \right\|_1$$

- Thus, for an arbitrary row of B, we have:

$$\left(1-\mu\lambda_i\right)^2 + \mu^2\lambda_i\left((\gamma-1)\lambda_i + \mathrm{trace}(\Lambda)\right) < 1$$

- From which we can follow for μ:

$$0 < \mu \leq \frac{2}{\gamma\lambda_{\max} + \mathrm{trace}(\Lambda)} \leq \frac{2}{\gamma\lambda_i + \mathrm{trace}(\Lambda)}$$

LMS Algorithm: Performance Measures

(parameter vector) error

system mismatch:

$$
\begin{aligned}
E\left[\underline{\tilde{\mathbf{w}}}_k^H \underline{\tilde{\mathbf{w}}}_k\right] &= \operatorname{trace}\left(E\left[\underline{\tilde{\mathbf{w}}}_k \underline{\tilde{\mathbf{w}}}_k^H\right]\right) \\
&= \operatorname{trace}\left(P_k\right) \\
&= \underline{1}^T \underline{c}_k = \sum_{l=1}^{M} \gamma_l \lambda_{B,l}^k
\end{aligned}
$$

Univ.-Prof. Dr.-Ing. Markus Rupp

LMS Algorithm: Performance Measures

Steady-state system mismatch:

$$\mathrm{trace}(P_k) = \sum_{l=1}^{M} \gamma_l \lambda_{B,l}^k$$

$$\begin{aligned}
\lim_{k\to\infty} \underline{c}_k &= \underline{c}_\infty = B\underline{c}_\infty + \mu^2 \underline{\lambda} \sigma_v^2 \\
&= [I - B]^{-1} \mu^2 \underline{\lambda} \sigma_v^2 \\
&= \left[2\Lambda - \mu\Lambda^2\gamma - \mu\underline{\lambda}\underline{\lambda}^T\right]^{-1} \mu\underline{\lambda}\sigma_v^2
\end{aligned}$$

Univ.-Prof. Dr.-Ing. Markus Rupp

LMS Algorithm: Performance Measures

Matrix-Inversion-Lemma

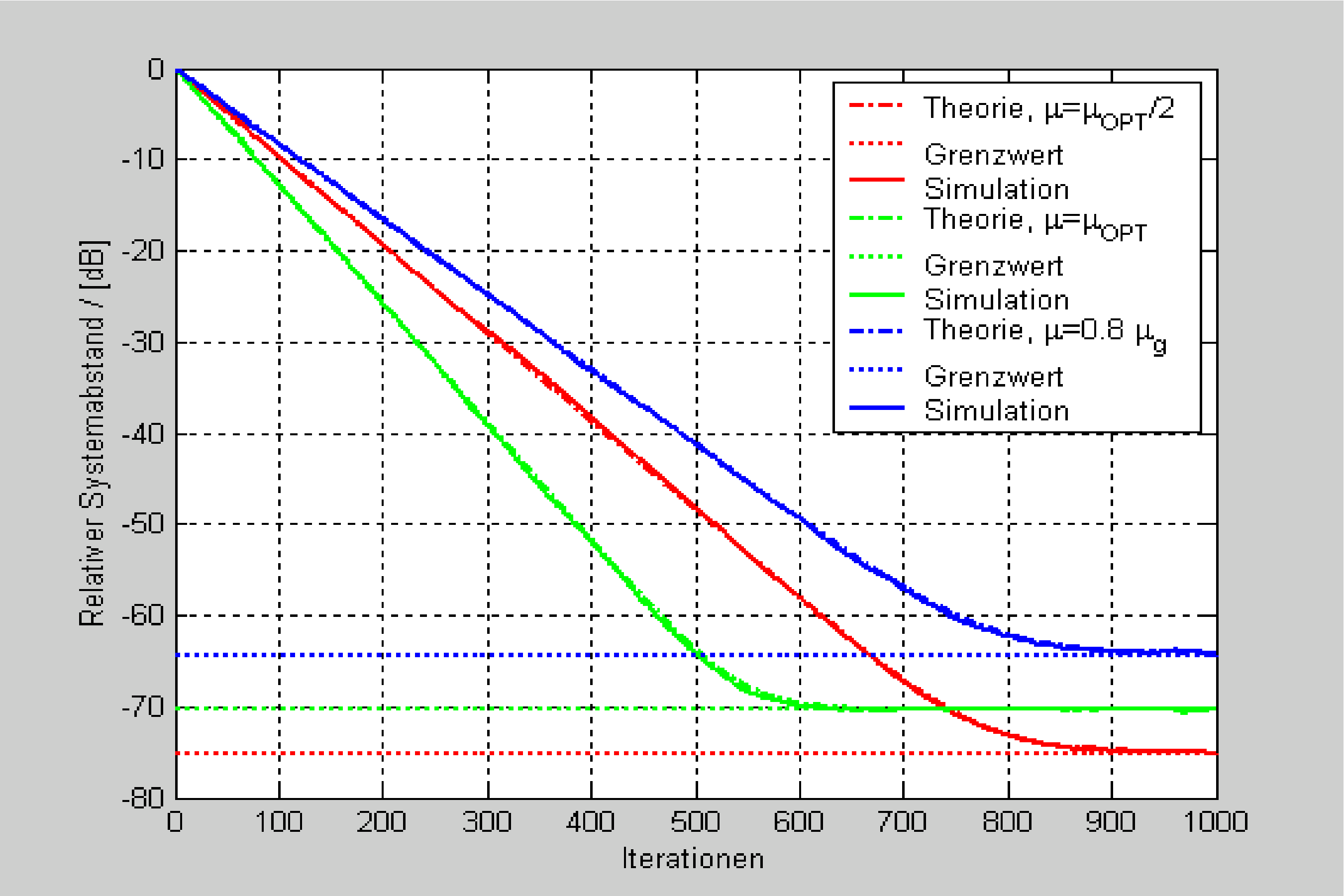
$$\underline{1}^T \underline{c}_\infty = \mu\sigma_{\mathbf{v}}^2 \frac{\sum_{l=1}^{M} \frac{1}{2-\mu\gamma\lambda_l}}{1-\mu\sum_{l=1}^{M}\frac{\lambda_l}{2-\mu\gamma\lambda_l}}$$

Small step-sizes μ

$$\underline{1}^T \underline{c}_\infty = \frac{\mu\sigma_{\mathbf{v}}^2 M}{2}$$

Univ.-Prof. Dr.-Ing. Markus Rupp

LMS Algorithm: Performance Measures



Univ.-Prof. Dr.-Ing. Markus Rupp

LMS Algorithm: Performance Measures

Relative system mismatch

$$\frac{\underline{1}^T \underline{c}_\infty}{\|\underline{w}_o\|}$$

Distorted a-priori error:

$$\tilde{\mathbf{e}}_{a,k} = \mathbf{d}_k - \underline{\mathbf{x}}_k^T \underline{\hat{\mathbf{w}}}_{k-1}$$

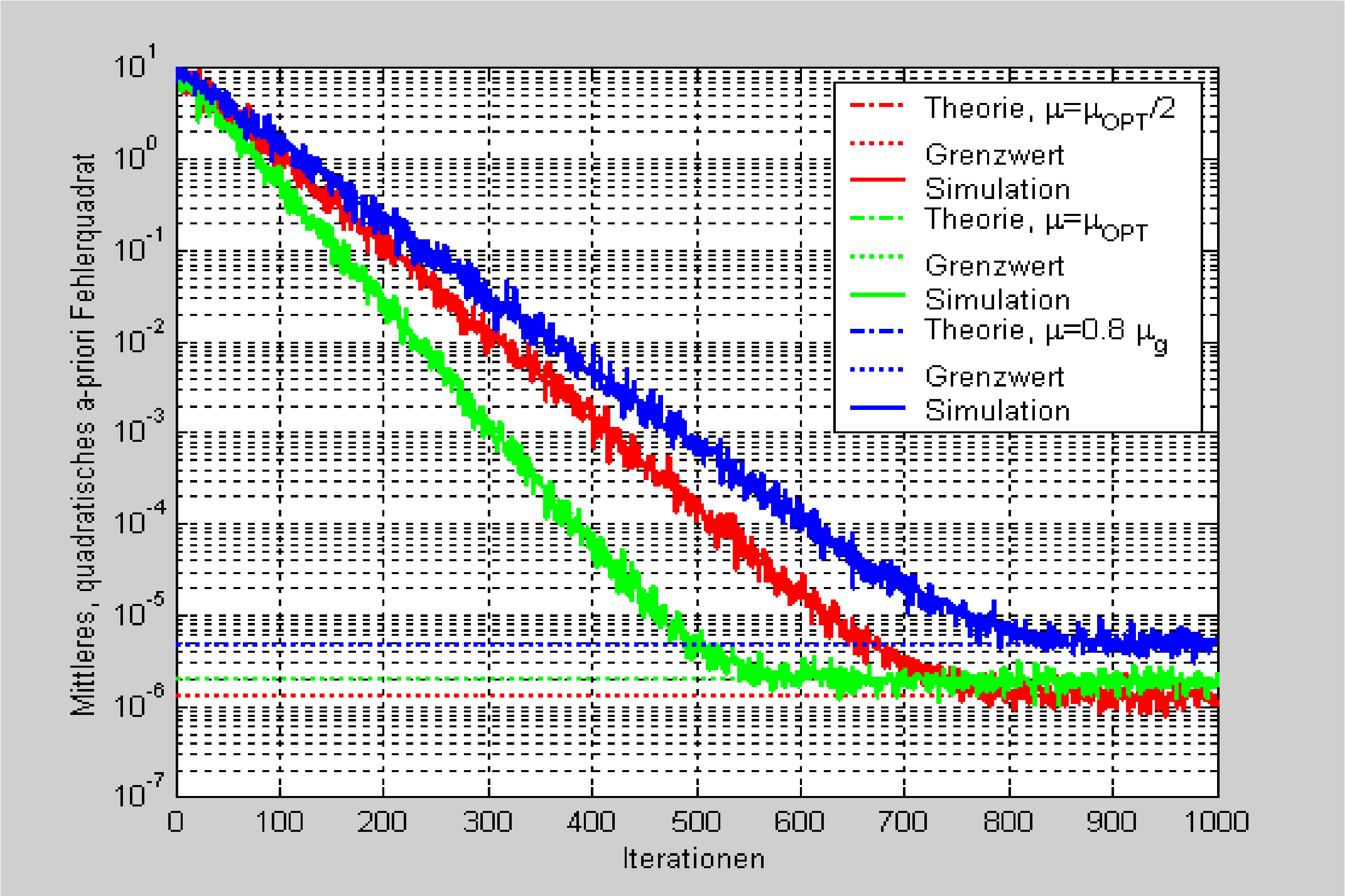
Univ.-Prof. Dr.-Ing. Markus Rupp

LMS Algorithm: Performance Measures

$$
\begin{aligned}
\mathrm{E}\left[\left|\tilde{\mathbf{e}}_{a,k}\right|^2\right] &= \mathrm{E}\left[\left|\mathbf{d}_k - \underline{\hat{\mathbf{w}}}_{k-1}^T \underline{\mathbf{x}}_k\right|^2\right] \\
&= \mathrm{E}\left[\left|\mathbf{v}_k - \underline{\tilde{\mathbf{w}}}_{k-1}^T \underline{\mathbf{x}}_k\right|^2\right] \\
&= \sigma_{\mathbf{v}}^2 + \mathrm{E}\left[\left|\underline{\tilde{\mathbf{w}}}_{k-1}^T \underline{\mathbf{x}}_k\right|^2\right] \\
&= \sigma_{\mathbf{v}}^2 + \mathrm{E}\left[\underline{\tilde{\mathbf{w}}}_{k-1}^T \underline{\mathbf{x}}_k \underline{\mathbf{x}}_k^H \underline{\tilde{\mathbf{w}}}_{k-1}^*\right] \\
&= \sigma_{\mathbf{v}}^2 + \mathrm{E}\left[\underline{\tilde{\mathbf{w}}}_{k-1}^T R_{\underline{\mathbf{xx}}}^* \underline{\tilde{\mathbf{w}}}_{k-1}^*\right] \\
&= \sigma_{\mathbf{v}}^2 + \mathrm{trace}\left(\mathrm{E}\left[R_{\underline{\mathbf{xx}}}^* \underline{\tilde{\mathbf{w}}}_{k-1}^* \underline{\tilde{\mathbf{w}}}_{k-1}^T\right]\right) \\
&= \sigma_{\mathbf{v}}^2 + \underline{\lambda}^T \underline{c}_{k-1}
\end{aligned}
$$

Univ.-Prof. Dr.-Ing. Markus Rupp

LMS Algorithm: Performance Measures



Univ.-Prof. Dr.-Ing. Markus Rupp

LMS Algorithm: Performance Measures

Excess Mean Square Error

$$g_{\mathrm{ex}} = \underline{\lambda}^T \underline{c}_\infty = \mu \sigma_{\mathbf{v}}^2 \frac{\sum_{l=1}^{M} \frac{\lambda_l}{2-\mu\gamma\lambda_l}}{1-\mu \sum_{l=1}^{M} \frac{\lambda_l}{2-\mu\gamma\lambda_l}}$$

Misadjustment

$$m_{\mathrm{LMS}} = \frac{g_{\mathrm{ex}}}{g_o} = \mu \frac{\sum_{l=1}^{M} \frac{\lambda_l}{2-\mu\gamma\lambda_l}}{1-\mu \sum_{l=1}^{M} \frac{\lambda_l}{2-\mu\gamma\lambda_l}}$$

Univ.-Prof. Dr.-Ing. Markus Rupp

Learning curves

- Note that there is some subtleties in averaging learning curves.

- If the outcome of an experiment is $||\underline{\mathbf{w}}-\underline{\mathbf{w}}_k||_2^2$ then averaging over N experiments is straightforward:
  $1/N \, \Sigma||\underline{\mathbf{w}}-\underline{\mathbf{w}}_k||_2^2$

Learning curves

- However, how do we average
  $||\underline{\mathbf{w}}-\underline{\mathbf{w}}_k||_2^2 \,/\, ||\underline{\mathbf{w}}||_2^2$
- The random variable $\underline{\mathbf{w}}$ now appears in numerator and denominator!
- Jensens inequality: for convex functions:
- (for concave the other way around)

$$f\left(E[x]\right) \leq E\left[f\left(x\right)\right]$$

Learning curves

- Therefore, we have to do the following:
- $Nu = 1/N \, \Sigma||\underline{\mathbf{w}}-\underline{\mathbf{w}}_k||_2^2$
- $De = 1/N \, \Sigma||\underline{\mathbf{w}}||_2^2$
- $Sys_{rel}(k) = Nu/De$