Adaptive Systems

Assignment 2

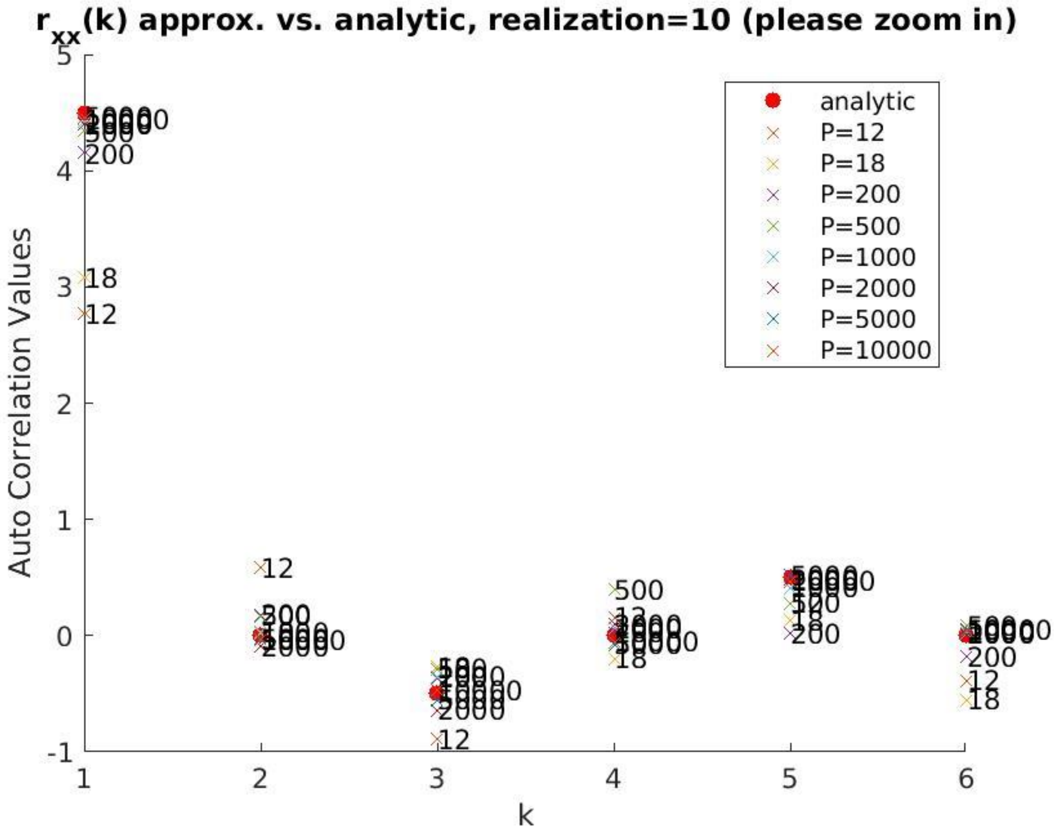
Thanks Ziad and Thomas for your help again!

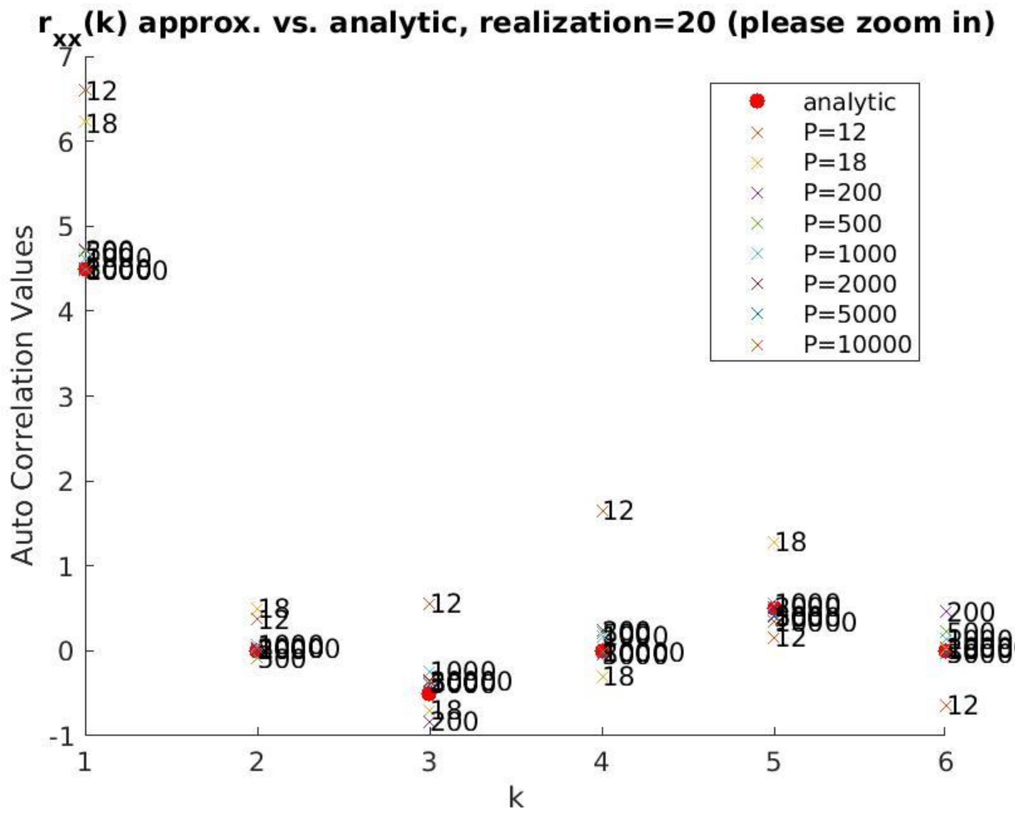
Task 2.2c

Please use the deployed matlab script (“task_2_2_bc.m”) for investigating correct computation. It supports three functions “compare_rxx”, “compare_p_rdx”, “compare_h_c” (optional). “task_2_2_bc.m” requires additional matlab file “r_dx.m”

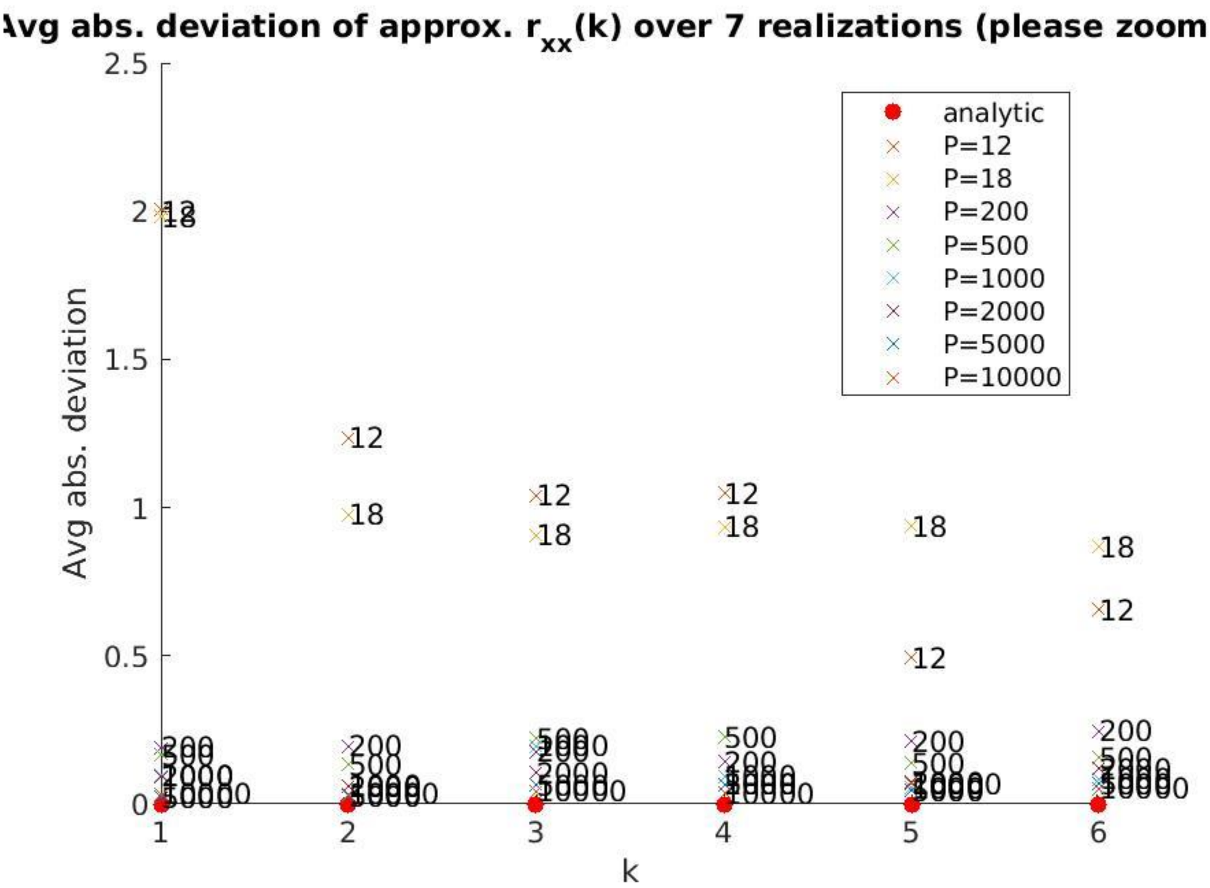
Matlab supports interactive “zoom in” in figures. This feature is necessary, because the data points are very close together, impossible to discriminate them in the figures below. Here only two example figures are shown as a kind of “concept proof”. Matlab script “task_2_2_bc.m” plots r_xx(k) and p(k) for realization=10,20,30,40,50,80,100. The plots visualize the difference of the approximated r_xx(k) / p(k) vs their analytic values. Additionally the “average abs. deviation” over all used realizations is plotted below, separately for rxx(k) and p(k).

Approximation of r_xx(k)





Taking the average of the absolute deviations over all used realizations (10,20…) results in the following figure ("average of figures above"):



Discussion r_xx(k):

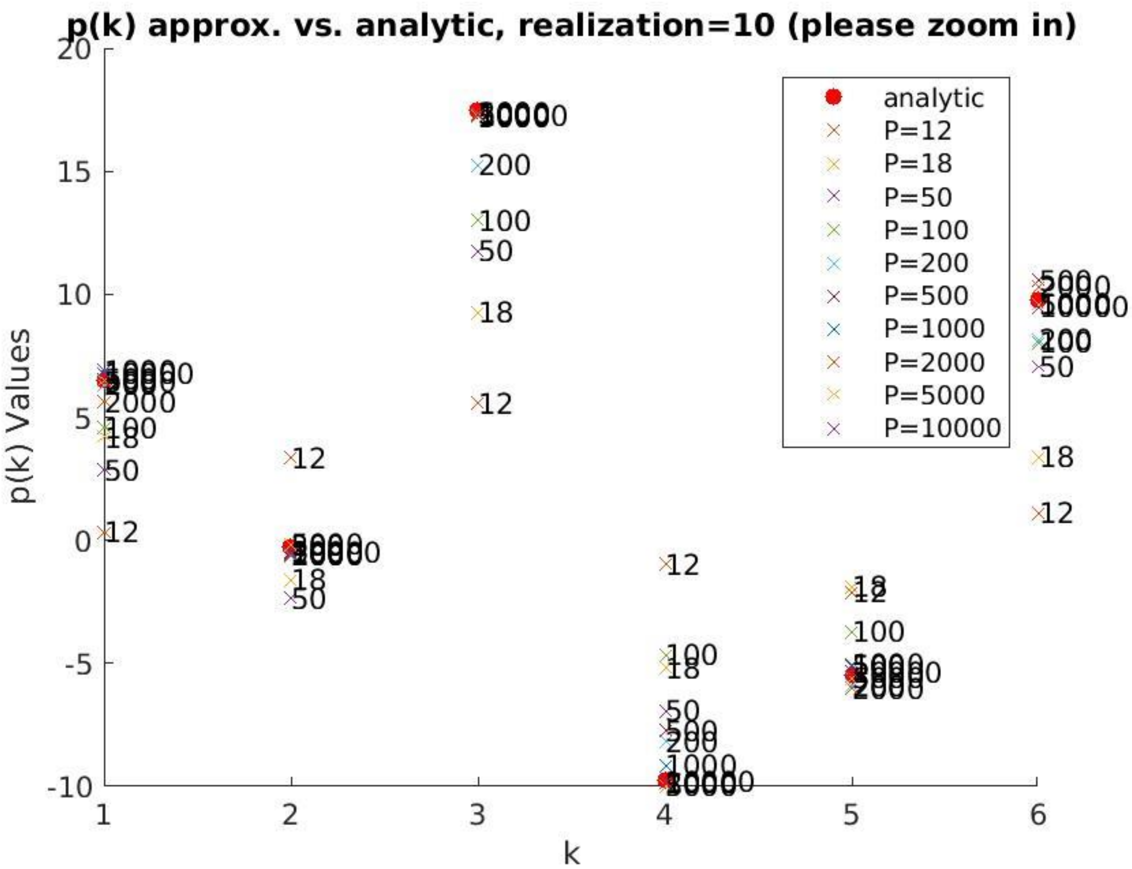
Diamond shape of the discrete auto correlation (not shown here) approximation remains roughly the same for different P (>1000) (total length=2*P-1)

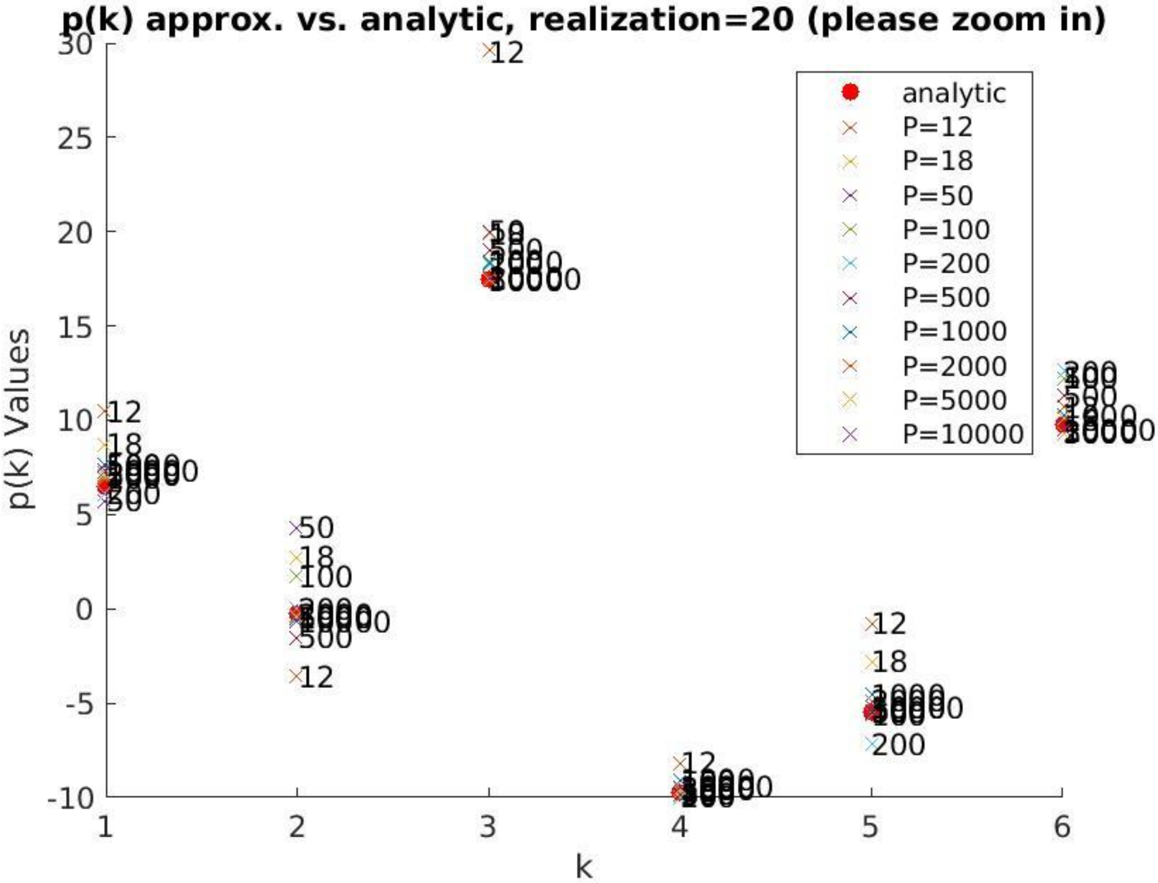
The innermost, let's say, 10% of the autocorrelation values located at the center ("k=0") may be used as approximations. In absolute numbers, these 10% result in bigger numbers for big Ps than for low Ps, therefore more coefficients can be approximated finally by Wiener Hopf, when using big Ps. The number of useful autocorrelation values specify the autocorrelation matrix dimension, which in turn is related to the number of filter coefficients.

The plots clearly show better approximations for rxx(k)/p(k) for bigger Ps on the average (varies a bit from one realization to another one, but please take a look at figure “Avg abs. dev…” above and for p(k) below).

P<2000 seems to work rather poor (but depends on number of wanted coefficients, maximum allowed error, application scenario,...)

Approximation of p(k)

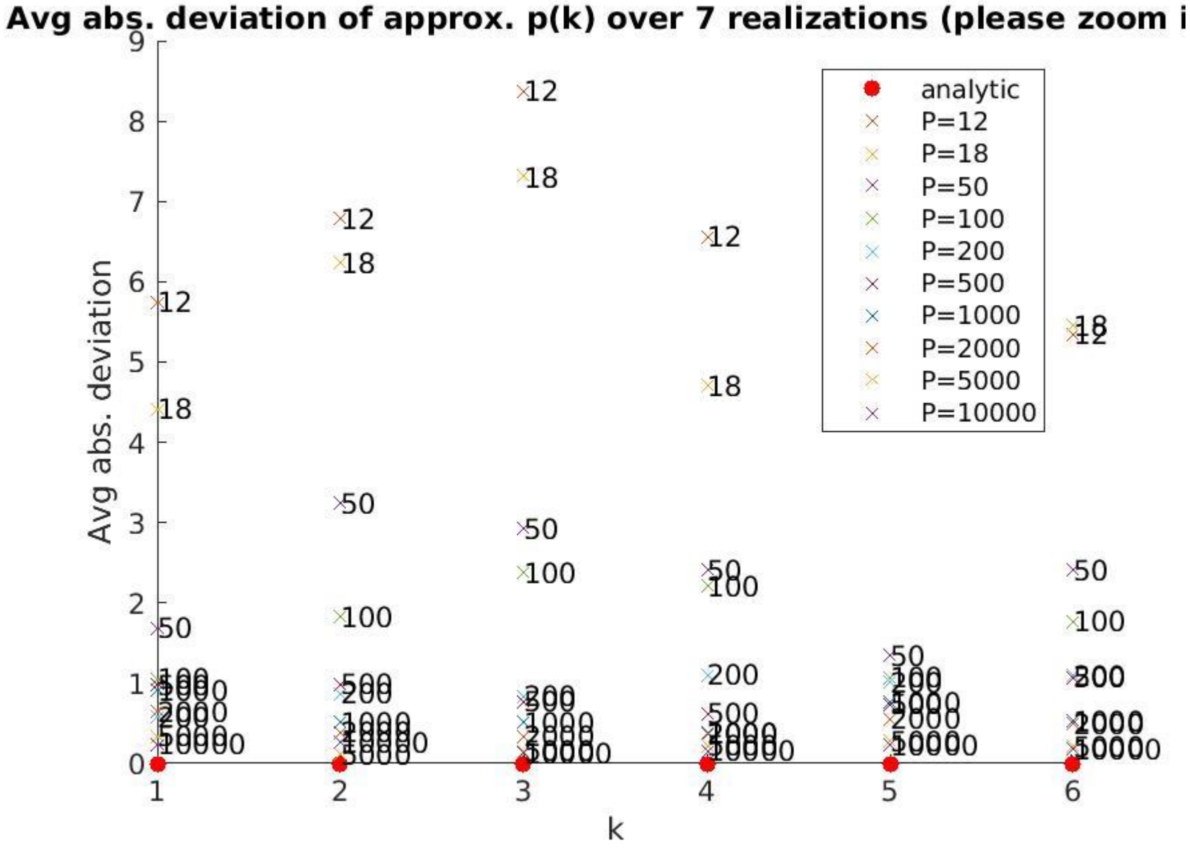




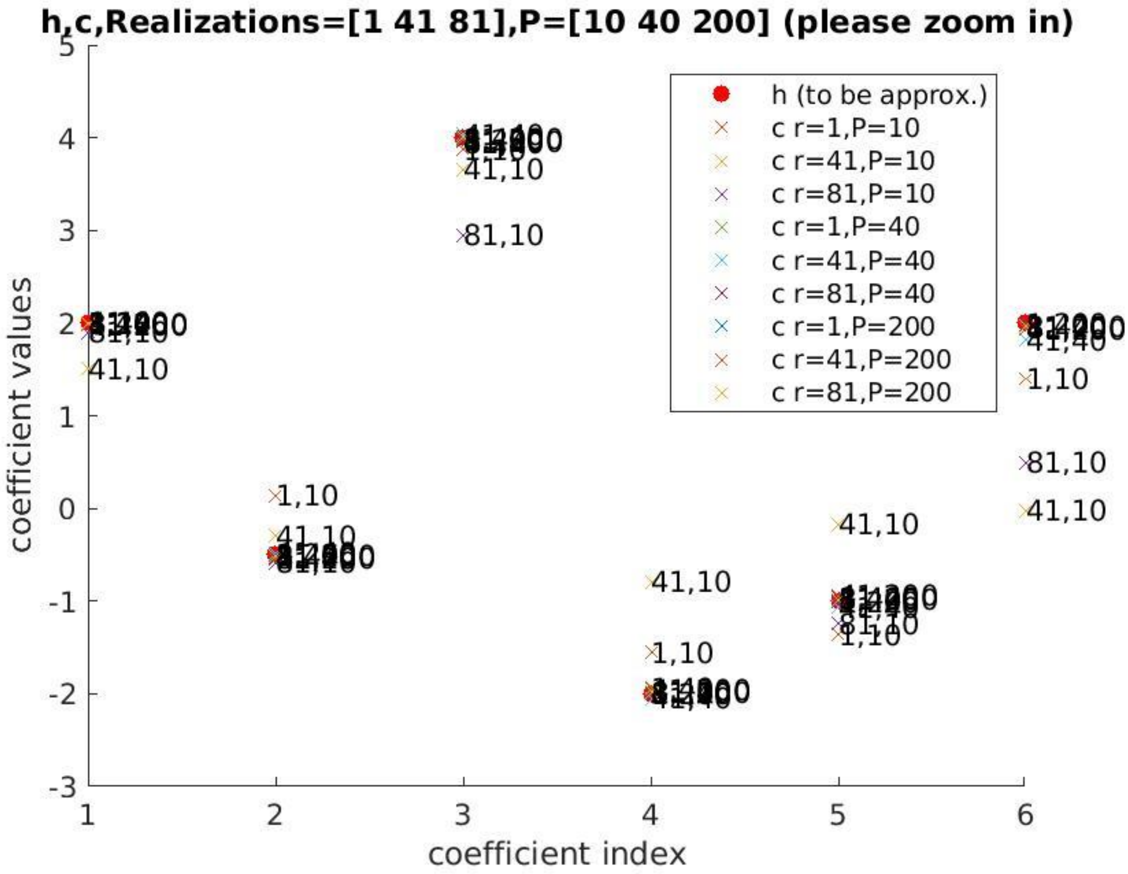
Discussion of p(k)

The bigger P, the better the approximation. But some deviations remain even for P=10000. Although no error criterion given, P>=2000 seems to be necessary (for most realizations).

Finally the average absolute deviation of the approximated p(k) over 7 realizations is shown again:


Avg abs. deviation of approx. p(k) over 7 realizations (please zoom i
Avg abs. deviation
k
analytic
P=12
P=18
P=50
P=100
P=200
P=500
P=1000
P=2000
P=5000
P=10000

Optional comparison of h and c:



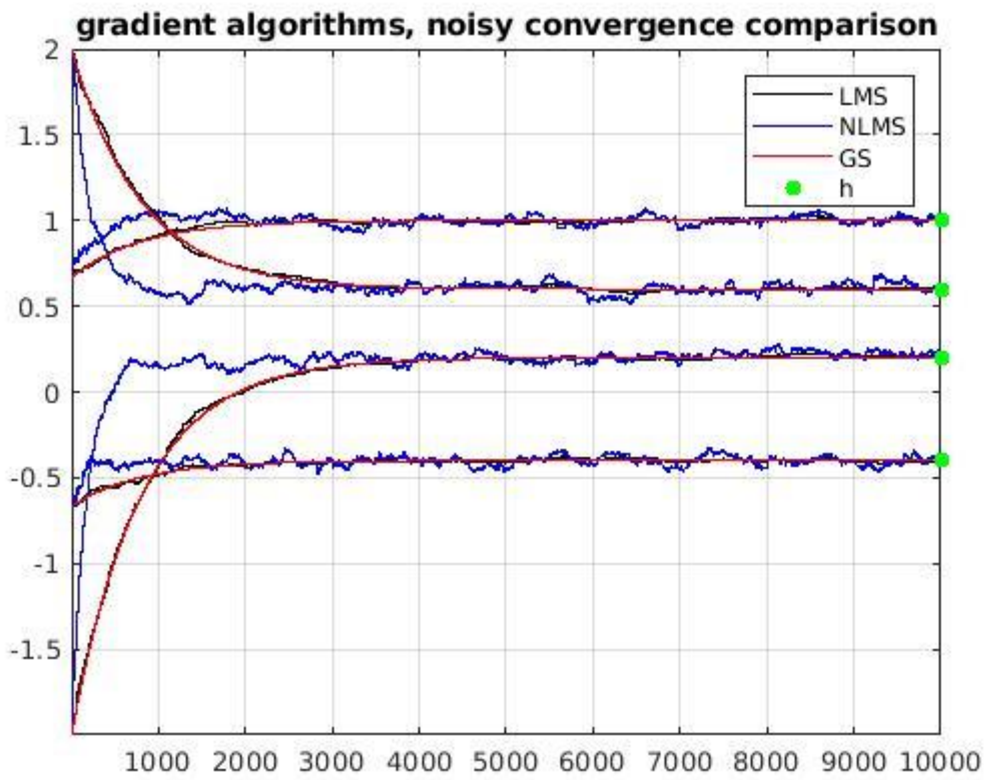
Discussion

No error criterion with respect to maximal deviation of the approximated coefficient is specified, but P=40 seems to deliver good approximations (~at most realizations). P=200 only slight deviations any more (regardless of used realization).

Task 2.4

a and b) Matlab code is contained in submitted package.

c) Fourth plot shows "convergence path" of h's coefficients for different gradient algorithms. GD goes along a smooth path (because it is following it's pre-estimated signal statistics, which remains constant), whereas LMS and NLMS fluctuate on its way to approximate h. LMS and NLMS depend on x(n) causing the algorithms to "follow" the current fluctuating input signal value while trying to find optimum approximation. h is hit precisely by all of them. The following figure is based on a non-coefficient-leakage implementation of "gd_algorithm" and "lms_algorithm" (or alpha has to be set to 0 otherwise, see task 2.5e).

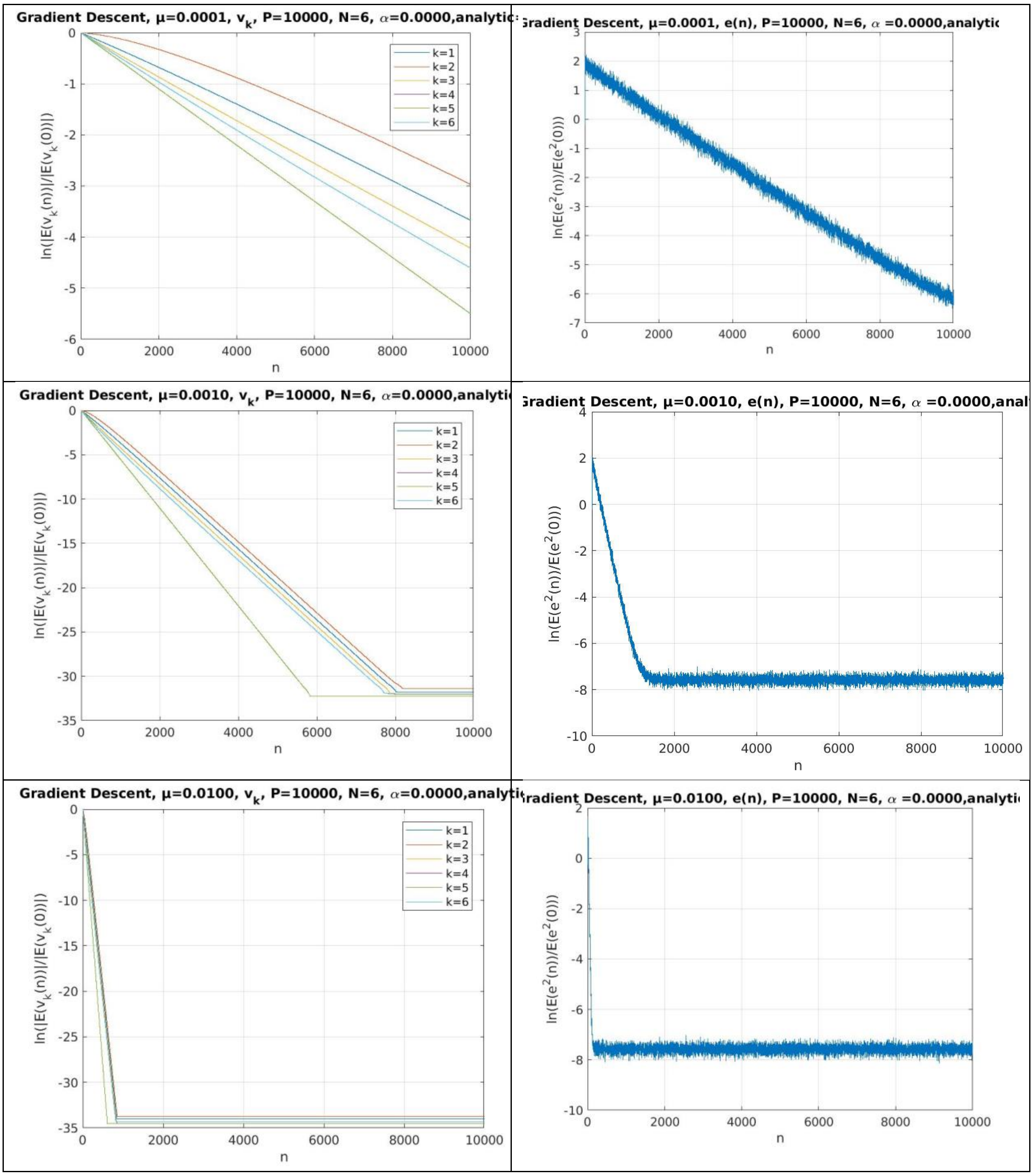


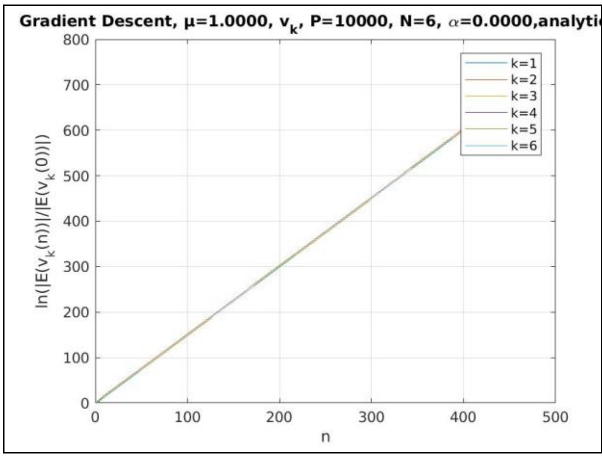
Note: Submitted matlab code for "gd_algorithm" and "lms_algorithm" implement coefficient leakage (task 2.5.e). If 'h' is not hit precisely by a test run of "gradient_test.m", the reason will be the alpha value, which is set at fourth figure to 0.1!

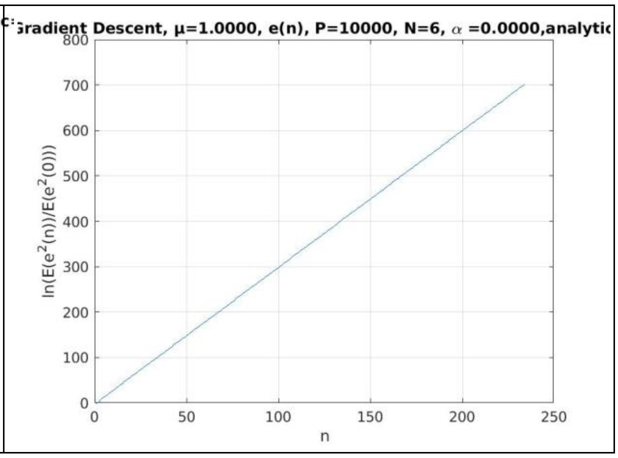
Task 2.5a

Matlab code generates two plots for each μ={0.0001, 0.001, 0.01,1}. For in-depth inspection please use deployed matlab script “task_2_5.m” and activate the corresponding task’s function (“task_2_5_a()”). Plots provided here may serve only as “reference to matlab”.

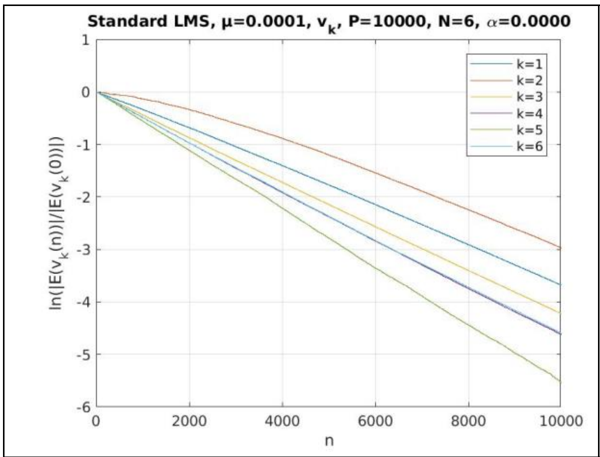
Gradient Descent

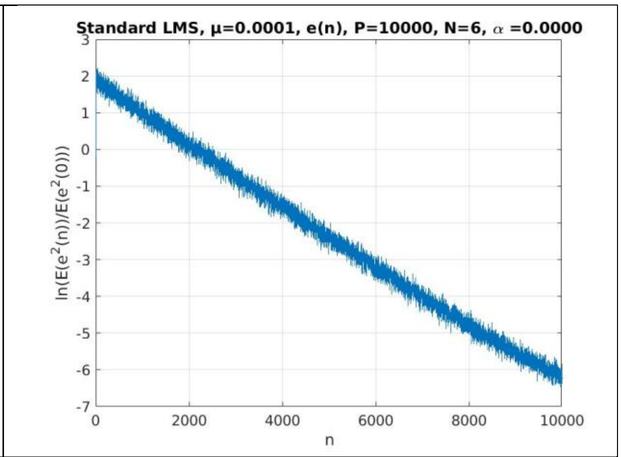


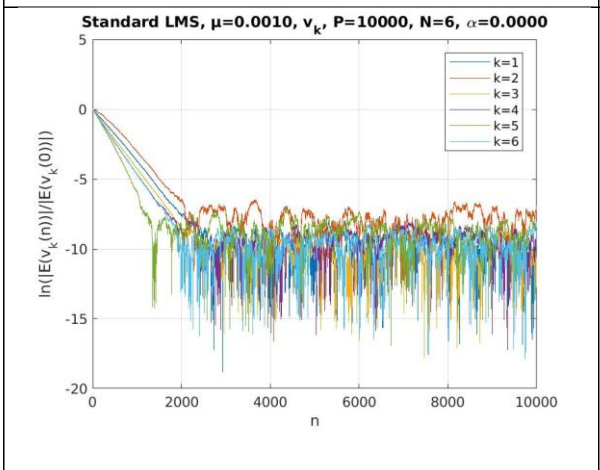


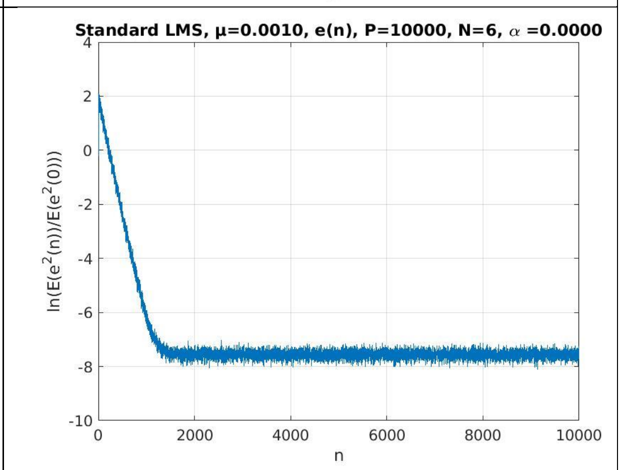


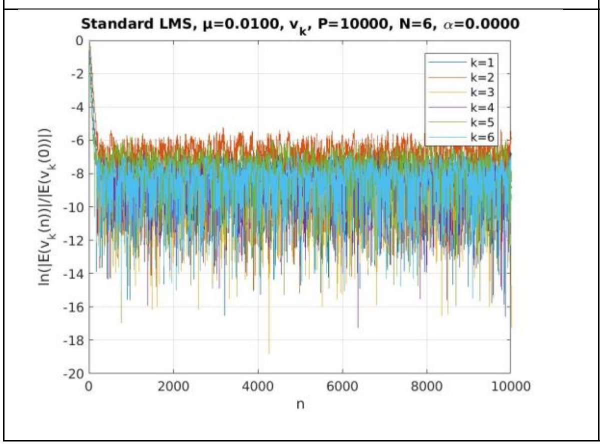
Standard LMS

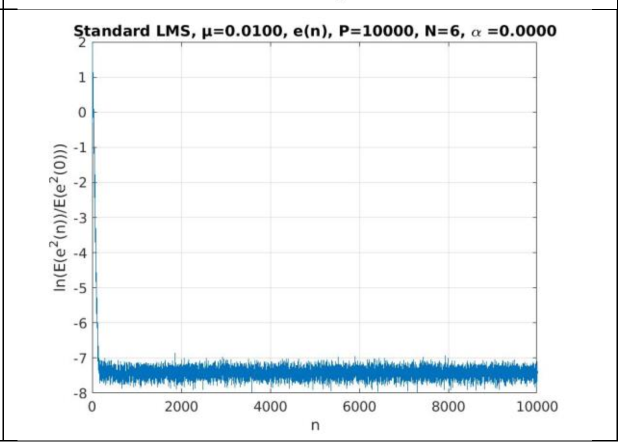


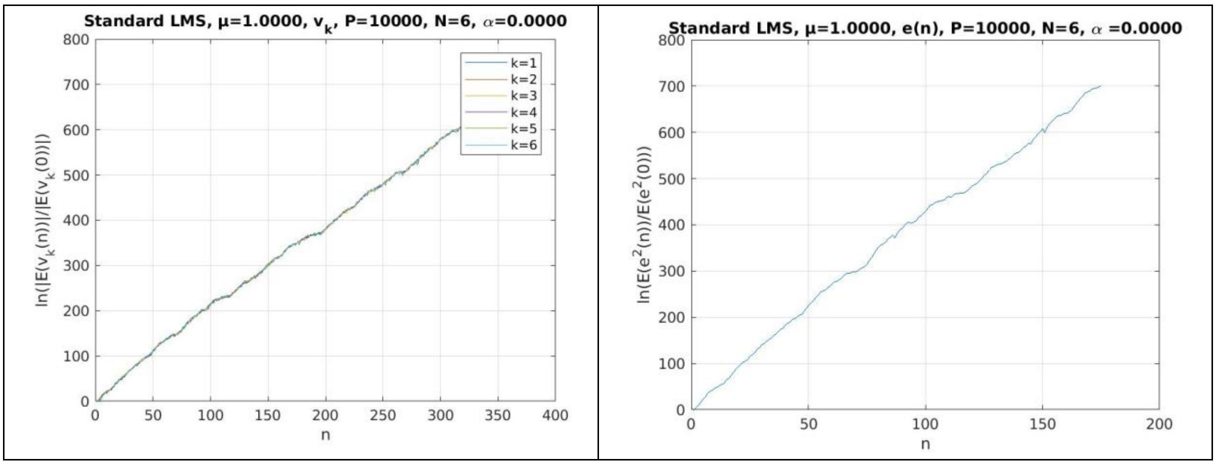












Discussion

μ=0.0001: Both LMS and GD are stable, but need more time (>10000 samples) to converge.

μ=0.001: Stable, converges within 10000 samples

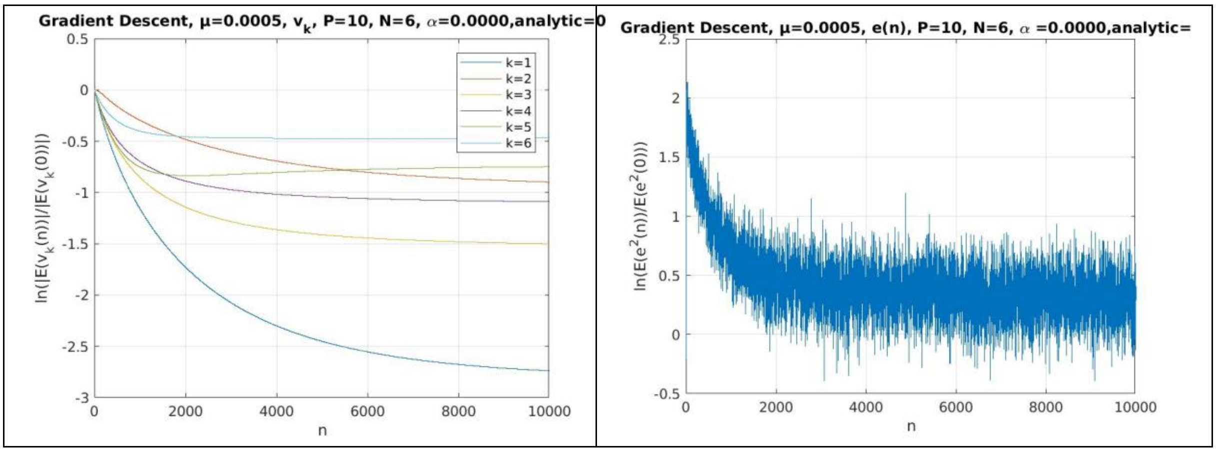
μ=0.01: Stable, converges faster than μ=0.001

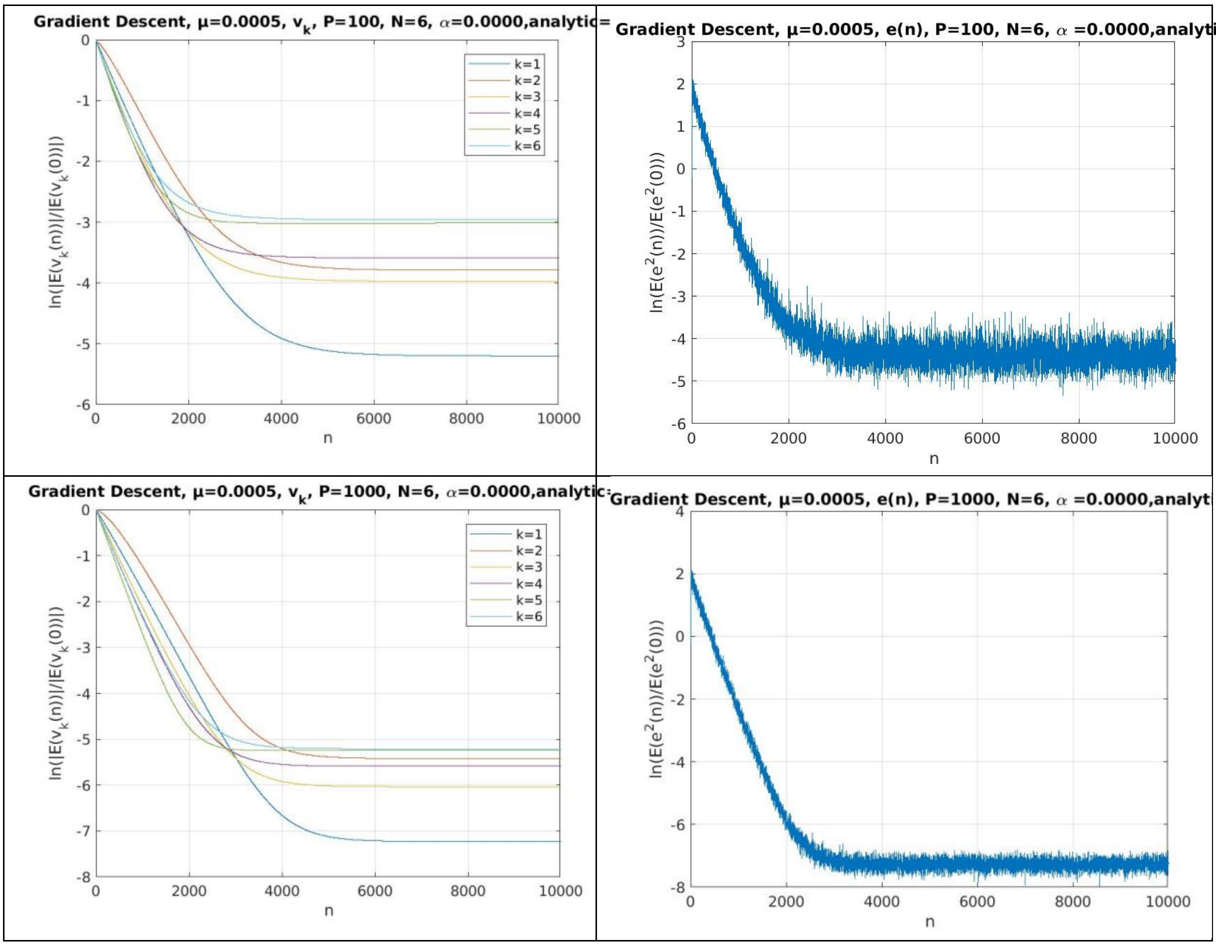
μ=1 -> Neither LMS nor GD are stable, error plot terminates after a few hundred samples due to coefficient values' overflow (->NaN)

Task 2.5b

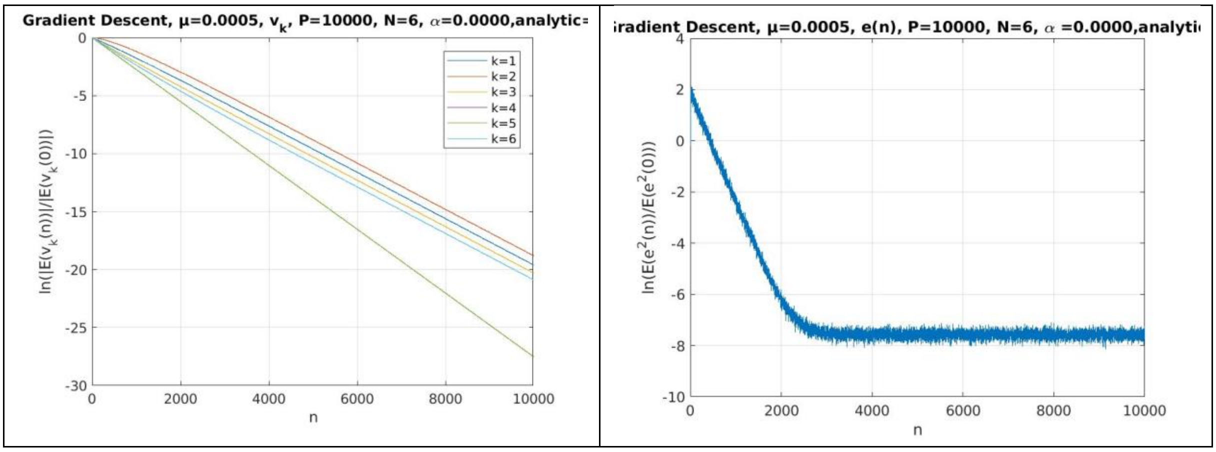
Gradient Descent variation over P={10,100,1000}. }. For in-depth inspection please use deployed matlab script “task_2_5.m” and activate the corresponding task’s function (“task_2_5_b()”). Plots provided here may serve only as “reference to matlab”.

Gradient Descent using estimates for Rxx,p





Gradient Descent using analytic values for Rxx,p



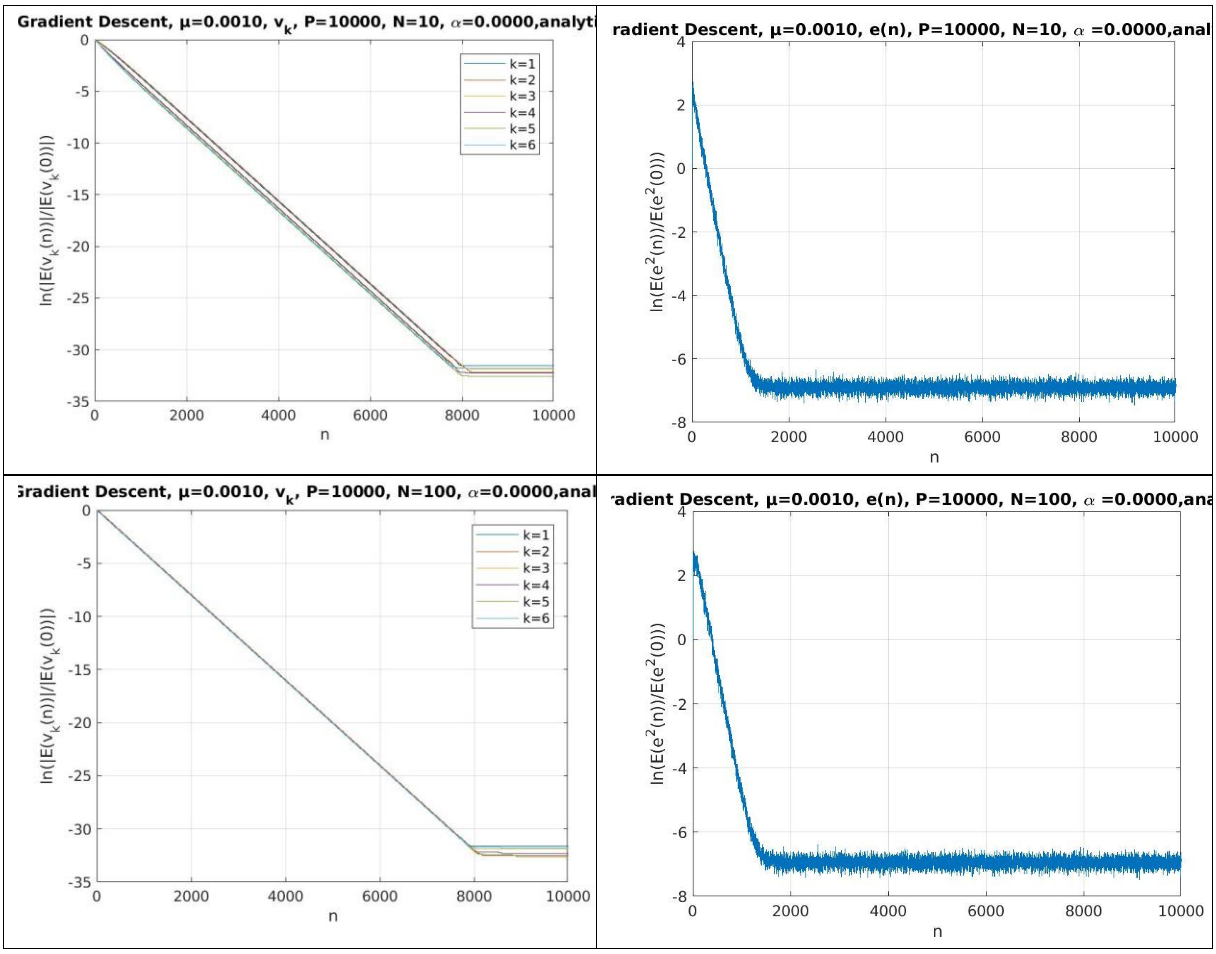
Discussion

- The higher P, the better the convergence (and the later the point of convergence is achieved but much, much earlier than the analytic one).
- The higher P, the lower the ln(E_v_k_ratio) and ln(E_e^2_ratio) once convergence has been achieved.
- The higher P, the smoother ln(E_e^2_ratio). ln(E_e^2_ratio)). For P=1000 close to analytic ln(E_e^2_ratio)).

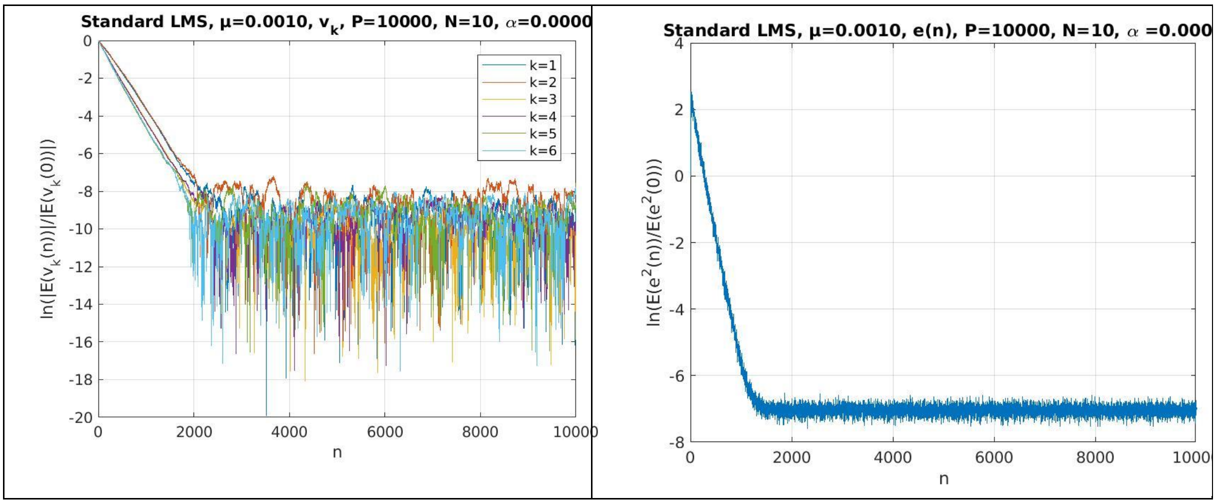
Task 2.5c

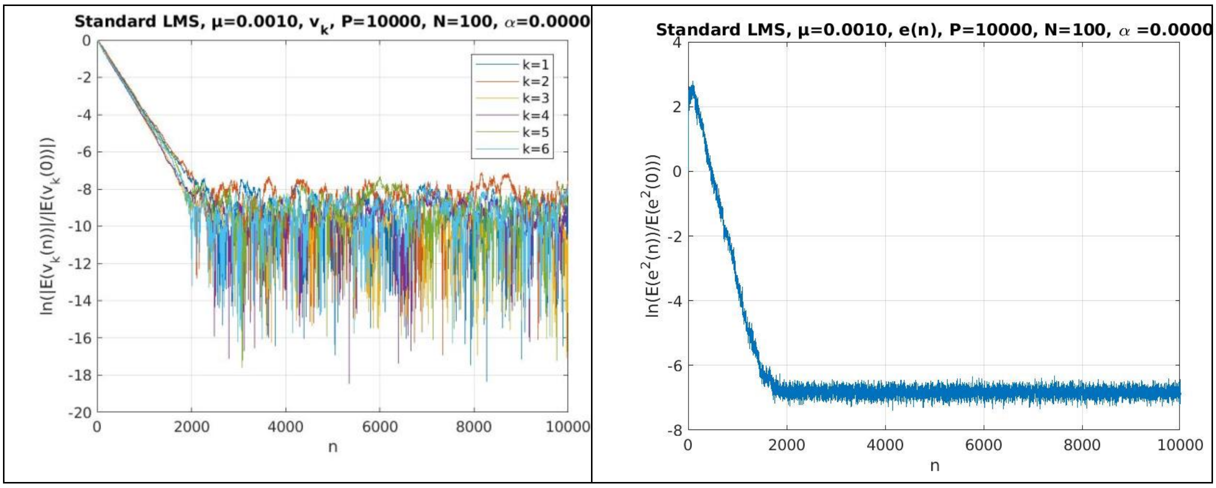
Gradient Descent and LMS variation over N={10,100}. For in-depth inspection please use submitted matlab script “task_2_5.m” and activate the corresponding task’s function (“task_2_5_c()”). Plots provided here may serve only as “reference to matlab”.

Gradient Descent



Standard LMS





Discussion

- On a coarse level neither the LMS nor the GD seem to benefit dramatically from a large N, both ln(E_v_ratio) and ln(E_e^2_ratio) seem to be similar (apart from some minor differences described below).
- On closer inspection (plot ln(E_v_k_ratio)) the individual error ratios during convergence for the v_k-coefficients are more similar for N=100 than for N=10 (the convergence paths are closer together).
- For N=10,100 h is zero padded at the end (size("native" h)=6), these extra entries will differ at startup from randomly initialized c0, finally causing an increase at beginning of ln(E_e^2_ratio). Therefore it also takes some extra convergence time to compensate this additional error (ln(E_e^2_ratio) "error shape" is shifted to the right for N=100 compared to N=10). Once converged the ln(E_e^2_ratio) is also a bit higher (=worse) for N=100 than for N=10 (both GD and LMS). In order to observe this effect it may be necessary to use matlab, upper image quality will be too low.
- Required computational power increases too!
- Finally, it depends on the application scenario, if the increase in required computational power is worth to get only slightly better results in terms of ln(E_e^2_ratio).

Task 2.5d

"MSE behaviour after convergence"

LMS update rule c(n)=c(n-1)+μ*e(n)*x(n)

As can be seen in the formula the LMS depends on x(n) causing fluctuations in c's convergence path (also visible in ln(E_v_k_ratio)). Once converged the LMS follows "new" input data in order to optimize the coefficients furtheron continuously. This causes fluctuations in the LMS ln(E_v_ratio) figures. Fluctuations will vary from one realization to the next.

In contrast the GD update rule: c(n)=c(n-1)+μ(p-Rxx*c(n-1))

GD update rule does not depend on x(n) rather on it's statistical estimations Rxx/p, which are computed beforehand (either analytically or estimated for a longer time interval) and remain constant causing finally a smooth convergence path. The convergence path is much smoother (a line) than the LMS' one, both during and after convergence (for all coefficients). This behaviour is implicit to the algorithm and remains over all signal realizations.

Task 2.5e

Coefficient leakage of GD and standard LMS

Where do they converge to?

For coefficient-leakage LMS the converged coefficients are scaled at infinity by (sigma_x^2/(alpha+sigma_x^2)) (analytically computed at task 2.3.a).

Example, using provided “gradient_test.m” of task 2.4: sigma_x^2=0.25, alpha=0.1

If 'h' is scaled by 0.3846 (from above formula), these scaled values may be additionally scatter-plotted to figure “4” of task 2.4 as 'convergence goals' (coloured blue, please take a look at figure below). Figure 4 shows that these scaled h coefficients are hit precisely by LMS and GD.

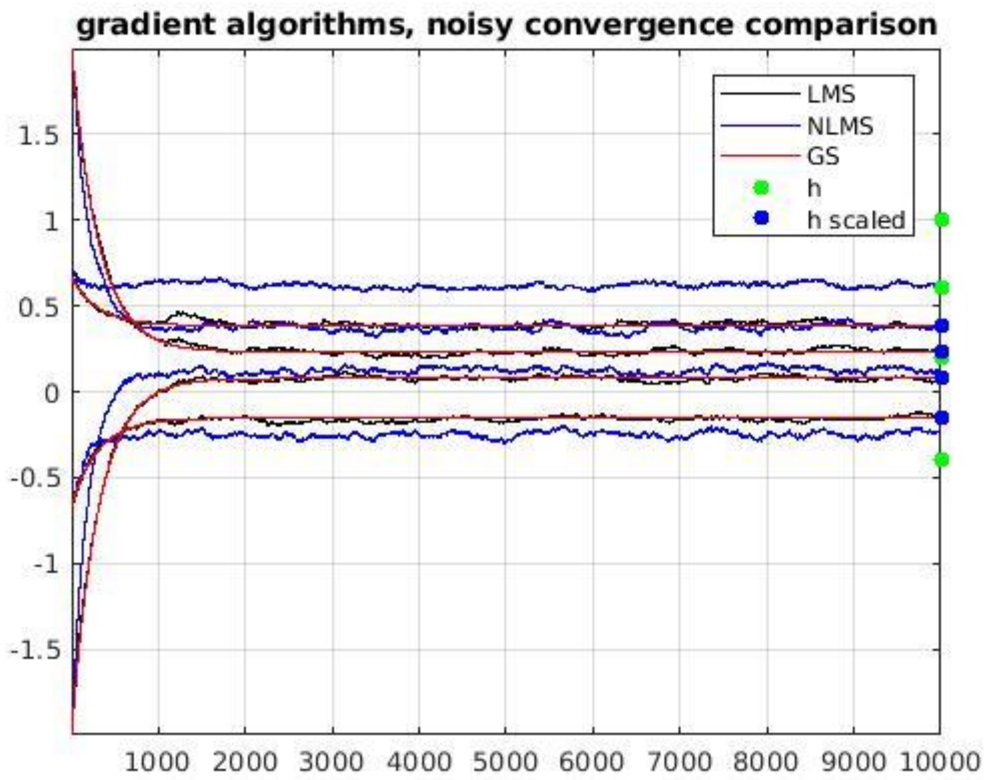
Submitted matlab code for “gd_algorithm” and “lms_algorithm” implement coefficient leakage.

Additional matlab code to extend figure 4 (to be inserted at line 140 in gradient_test.m):

```
scale_factor=sigma_x^2/(alpha+sigma_x^2);

scaled_h=h*scale_factor;

htrue_scaled=scatter(Nx*ones(Nh,1),scaled_h,'b','o','filled');
```



„Figure 4“

Assignment 2

2.1.a)

$$\underline{c}(n) = \underline{c}(n-1) + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1))$$

$$\underline{c}(n) = (1-\mu\alpha)\cdot\underline{c}(n-1) + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1))$$

$$J(\underline{c}(n)) = E(e^2(n)) + \alpha\,\|\underline{c}(n)\|^2 \qquad \|\underline{c}\|^2 = \underline{c}^T\cdot\underline{c}$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu\cdot\nabla_{\underline{c}} J(\underline{c})\Big|_{\underline{c}=\underline{c}(n-1)}$$

$$J(\underline{c}(n)) = E(e^2(n)) + \alpha\cdot\|\underline{c}(n)\|^2\Big|_{\underline{c}=\underline{c}(n-1)} \qquad e(n) = d(n) - \underline{c}^T(n-1)\cdot\underline{x}(n)$$

$$= E\left((d(n) - \underline{c}^T(n-1)\underline{x}(n))^2\right) + \alpha\,\|\underline{c}(n)\|^2\Big|_{\underline{c}=\underline{c}(n-1)}$$

$$= E\left(d^2(n) - 2d(n)\cdot\underline{c}^T(n-1)\cdot\underline{x}(n) + \underline{c}^T(n-1)\underline{x}(n)\underline{c}^T(n-1)\underline{x}(n)\right) + \alpha\,\|\underline{c}(n)\|^2\Big|_{\underline{c}=\underline{c}(n-1)}$$

$$= E(d^2(n)) - 2\underline{c}^T(n-1)E(d(n)\underline{x}(n)) + \underline{c}^T(n-1)E(\underline{x}(n)\cdot\underline{x}^T(n))\cdot\underline{c}(n-1) + \alpha\,\|\underline{c}(n)\|^2\Big|_{\underline{c}=\underline{c}(n-1)}$$

$$= \sigma_d^2 - 2\underline{c}^T(n-1)\cdot E(d(n)\underline{x}(n)) + \underline{c}^T(n-1)E(\underline{x}(n)\cdot\underline{x}^T(n))\cdot\underline{c}(n-1) + \alpha\cdot\sum_{i=0}^{N-1} c_i^2(n-1)$$

$$\left[\alpha\cdot\sum_{i=0}^{N-1} c_i^2(n-1) = \alpha\cdot\sum_{\substack{i=0\\i\neq j}}^{N-1} c_i^2(n-1) + \alpha\cdot \underbrace{c_j^2(n-1)}_{i=j} \rightarrow \frac{\partial}{\partial c_j} = \alpha\cdot 2\cdot c_j(n-1) \rightarrow \nabla_c(\alpha\|\underline{c}(n-1)\|^2) = 2\alpha\cdot\underline{c}(n-1)\right]$$

$$\nabla_{\underline{c}} J(\underline{c}) = -2\underline{p} + 2\underline{R}_{xx}\,\underline{c}(n-1) + 2\alpha\,\underline{c}(n-1)$$

$$\rightarrow \underline{c}(n) = \underline{c}(n-1) - \mu\cdot\nabla_{\underline{c}} J(\underline{c})$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu(-2\underline{p} + 2\underline{R}_{xx}\,\underline{c}(n-1) + 2\alpha\cdot\underline{c}(n-1))$$

$$\underline{c}(n) = \underline{c}(n-1)(1-2\mu\alpha) + \mu(2\underline{p} - 2\underline{R}_{xx}\,\underline{c}(n-1)) \qquad 2\mu = \tilde{\mu}$$

$$\underline{\underline{c}(n) = \underline{c}(n-1)(1-\tilde{\mu}\alpha) + \tilde{\mu}(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1))}$$

Assignment 2

2.1. b)

$\underline{c}(n) = (1-\mu\alpha)\,\underline{c}(n-1) + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1))$ $\qquad \underset{n\to\infty}{c(n) \to c_\infty}$

$\underline{c}(n) = \underline{c}(n-1) - \mu\alpha\,\underline{c}(n-1) + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1))$

$\cancel{\underline{c}_\infty} = \cancel{\underline{c}_\infty} - \mu\alpha\,\underline{c}_\infty + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}_\infty)$

$\cancel{\mu}\alpha\,\underline{c}_\infty = \cancel{\mu}(\underline{p} - \underline{R}_{xx}\,\underline{c}_\infty)$

$\alpha\,\underline{c}_\infty = \underline{p} - \underline{R}_{xx}\,\underline{c}_\infty$

$\alpha \cdot \underline{c}_\infty + \underline{R}_{xx}\,\underline{c}_\infty = \underline{p}$

$(\alpha \cdot \underline{I} + \underline{R}_{xx}) \cdot \underline{c}_\infty = \underline{p}$

$\underline{\underline{c}_\infty = (\alpha\,\underline{I} + \underline{R}_{xx})^{-1} \cdot \underline{p}}$

Assignment 2

2.1.c) $\underline{c}(n) = (1-\mu\alpha)\cdot\underline{c}(n-1) + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1)) \quad | -\underline{c}_\infty$

$$\underline{c}(n) - \underline{c}_\infty = (1-\mu\alpha)\cdot\underline{c}(n-1) + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1)) - \underline{c}_\infty$$

$$\underline{c}(n) - \underline{c}_\infty = \underline{c}(n-1) - \mu\alpha\,\underline{c}(n-1) + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1)) - \underline{c}_\infty \qquad \underline{c}(n-1) - \underline{c}_\infty = \underline{v}(n-1)$$

$$\underline{v}(n) = \underline{v}(n-1) + \mu(\underline{p} - \underline{R}_{xx}\,\underline{c}(n-1) - \alpha\cdot\underline{c}(n-1)) \qquad | \underline{p} = \alpha\cdot\underline{c}_\infty + \underline{R}_{xx}\underline{c}_\infty$$

$$\underline{v}(n) = \underline{v}(n-1) + \mu(\alpha\cdot\underline{c}_\infty + \underline{R}_{xx}\underline{c}_\infty - \underline{R}_{xx}\,\underline{c}(n-1) - \alpha\cdot\underline{c}(n-1))$$

$$= \underline{v}(n-1) + \mu(\alpha\,\underline{c}_\infty + \underline{R}_{xx}(\underline{c}_\infty - \underline{c}(n-1)) - \alpha\cdot\underline{c}(n-1))$$

$$= \underline{v}(n-1) + \mu(\alpha(\underbrace{\underline{c}_\infty - \underline{c}(n-1)}_{-\underline{v}(n-1)}) + \underline{R}_{xx}(\underbrace{\underline{c}_\infty - \underline{c}(n-1)}_{-\underline{v}(n-1)}))$$

$$\underline{v}(n) = \underline{v}(n-1) + \mu(-\alpha\cdot\underline{v}(n-1) - \underline{R}_{xx}\,\underline{v}(n-1))$$

$$= \underline{v}(n-1) - \mu\alpha\,\underline{v}(n-1) - \mu\,\underline{R}_{xx}\,\underline{v}(n-1)$$

$$\underline{v}(n) = (\underline{I} - \mu\alpha\cdot\underline{I} - \mu\cdot\underline{R}_{xx})\cdot\underline{v}(n-1)$$

Recursion

$$\underline{v}(n) = (\underline{I} - \mu\cdot\alpha\cdot\underline{I} - \mu\cdot\underline{R}_{xx})^n\cdot\underline{v}(0)$$

Assignment 2

2.1. d)
e)

$$\underline{R}_{xx} = \underline{Q}\,\underline{\Lambda}\,\underline{Q}^H$$

$$\underline{v}(n) = (\underline{I} - \mu\alpha\cdot\underline{I} - \mu\,\underline{R}_{xx})\cdot\underline{v}(n-1)$$

$$\underline{v}(n) = \underline{v}(n-1) - \mu\alpha\,\underline{v}(n-1) - \mu\cdot\underline{R}_{xx}\,\underline{v}(n-1)$$

$$= \underline{v}(n-1) - \mu\alpha\,\underline{v}(n-1) - \mu\cdot\underline{Q}\,\underline{\Lambda}\,\underline{Q}^H\underline{v}(n-1)$$

$$\underline{Q}^H\underline{v}(n) = \underline{Q}^H\underline{v}(n-1) - \mu\cdot\alpha\cdot\underline{Q}^H\underline{v}(n-1) - \mu\,\underbrace{\underline{Q}^H\underline{Q}}_{\underline{I}}\,\underline{\Lambda}\,\underline{Q}^H\underline{v}(n-1)$$

$$\tilde{\underline{v}}(n) = \tilde{\underline{v}}(n-1) - \mu\cdot\alpha\cdot\underbrace{\underline{Q}^H\underline{v}(n-1)}_{\tilde{\underline{v}}(n-1)} - \mu\cdot\underline{\Lambda}\,\tilde{\underline{v}}(n-1)$$

$$\tilde{\underline{v}}(n) = \tilde{\underline{v}}(n-1) - \mu\cdot\alpha\cdot\tilde{\underline{v}}(n-1) - \mu\cdot\underline{\Lambda}\,\tilde{\underline{v}}(n-1)$$

$$\underline{\tilde{\underline{v}}(n) = (\underline{I} - \mu\cdot\alpha\cdot\underline{I} - \mu\cdot\underline{\Lambda})\cdot\tilde{\underline{v}}(n-1)}$$

$$\underline{\tilde{\underline{v}}(n) = \underbrace{(\underline{I} - \mu\cdot\alpha\cdot\underline{I} - \mu\cdot\underline{\Lambda})^n}\cdot\tilde{\underline{v}}(0)}$$

$$\begin{pmatrix} \tilde{v}_0(n) \\ \vdots \\ \tilde{v}_{N-1}(n) \end{pmatrix} = \begin{pmatrix} (1-\mu\alpha-\lambda_0\mu)^n & \emptyset & \cdots & \emptyset \\ \emptyset & \ddots & & \vdots \\ \vdots & & (1-\mu\alpha-\lambda_i\mu)^n & \emptyset \\ \emptyset & \cdots & \emptyset & (1-\mu\alpha-\lambda_{N-1}\mu)^n \end{pmatrix} \begin{pmatrix} \tilde{v}_0(0) \\ \vdots \\ \vdots \\ \tilde{v}_{N-1}(0) \end{pmatrix}$$

each element of $\tilde{\underline{v}}(n)$ is decoupled, „$\tilde{v}_i(n) = f(\lambda_i, \tilde{v}_i(0))$"

$$\tilde{v}_i(n) = (1-\mu\alpha-\lambda_i\cdot\mu)^n\cdot\tilde{v}_i(0)$$

Assignment 2

2.1. f)

$$\tilde{v}_i(n) = (1-\mu\alpha-\mu\cdot\lambda_i)^n \cdot \tilde{v}_i(0)$$

$$|\tilde{v}_i(n)| = |\underbrace{(1-\alpha\mu-\mu\lambda_i)^n}_{e^{-\frac{n}{\tau_i}}}| \cdot |\tilde{v}_i(0)|$$

$$e^{-\frac{n}{\tau_i}} = |1-\alpha\mu-\mu\cdot\lambda_i|^n$$

$$\ln e^{-\frac{n}{\tau_i}} = \ln|1-\mu\alpha-\mu\cdot\lambda_i|^n$$

$$-\frac{\cancel{n}}{\tau_i}\cdot\cancel{\ln e} = \cancel{n}\cdot\ln|1-\mu\alpha-\mu\cdot\lambda_i|$$

$$\tau_i = -\frac{1}{\ln|1-\alpha\mu-\mu\cdot\lambda_i|}$$

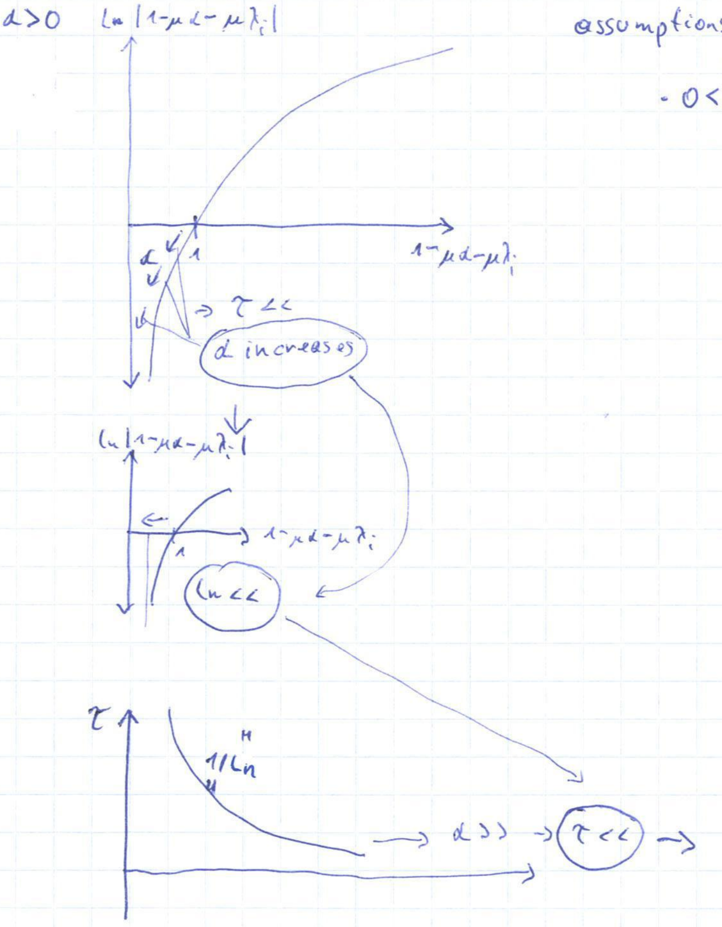
$$\tau_{max} = -\frac{1}{\ln|1-\mu\cdot\alpha-\mu\lambda_{min}|}$$

$$\tau_{min} = -\frac{1}{\ln|1-\mu\cdot\alpha-\mu\lambda_{max}|}$$

α influence : $\alpha=0 \Rightarrow$ „standard" gradient search

$\alpha>0$

assumptions: $\cdot\ \mu>0$
$\cdot\ \lambda>0$
$\cdot\ 0<|1-\mu\alpha-\mu\lambda|<1$



$\rightarrow$ α speeds up convergence (positive eigenvalues)

Assignment 2

2.1.f) α influence

1) $\mu > 0$
$\lambda > 0$

$\alpha \to 1$: $1 - \underbrace{\mu\alpha}_{>0} - \underbrace{\mu\lambda}_{>0} \to$ decreases $\to$ speedup

2) $\mu > 0$
$\lambda < 0$

$\alpha \to 1$: $1 - \mu\alpha - \mu\lambda$
$\underbrace{1 - \mu\lambda}_{>1} - \mu\alpha \to$ decreases $\to$ speedup BUT: depends on concrete values:
$1 - \mu\lambda - \mu\alpha < 1 \to$ decrease speedup
$1 - \mu\lambda - \mu\alpha > 1 \to$ decrease but $\tau < 0$ (no sense)

3) $\mu < 0$
$\lambda > 0$

$\alpha \to 1$: $1 - \mu\alpha - \mu\lambda \to$ increases $\to$ slows down BUT: $1 - \mu\alpha - \mu\lambda > 1 \to (n > 0 \to \tau < 0$ (no sense)

4) $\mu < 0$
$\lambda < 0$

$\alpha \to 1$: $1 - \mu\alpha - \mu\lambda \to$ increases $\to$ slows down BUT depends on: $|1 - \mu\alpha - \mu\lambda| > 1 \to \tau < 0$ (no sense)
$< 1 \to$ slows down

Assignment 2

2.2 a)

$$r_{xy}(k) = E(x(n+k)\cdot y(n)) = \lim_{P\to\infty} \frac{1}{P}\sum_{n=-P/2}^{P/2} x(n+k)\cdot y(n) = \frac{1}{P}\sum_{n=-P/2}^{+P/2} x(n+k)\cdot y(n) = \frac{1}{P}\sum_{n=0}^{P-1} x(n+k)\cdot y(n) = \hat{r}_{xy}(k)$$

↑ ergodicity

$$\mu_x = E(x) = \sum_{k=0}^{\infty} x_k(n)\cdot p_k = \lim_{P\to\infty}\frac{1}{P}\sum_{n=0}^{P-1} x(n) = \frac{1}{P}\cdot\sum_{n=0}^{P-1} x(n) = E(\hat{\mu}(P)) = E\left(\frac{1}{P}\sum_{n=0}^{P-1} x(n)\right) = \frac{1}{P}E\left(\sum_{n=0}^{P-1} x(n)\right) =$$

$$\frac{1}{P}\cdot\sum_{n=0}^{P-1} \underbrace{E(x(n))}_{\mu_x} = \frac{1}{P}\cdot P\cdot\mu_x = \mu_x$$

Assignment 2

2.3. a)

$$\underline{c}(n) = (1-\mu\alpha)\,\underline{c}(n-1) + \mu\, e(n)\cdot\underline{x}(n) \qquad e(n) = d(n) - y(n)$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu\alpha\cdot\underline{c}(n-1) + \mu\left(d(n) - \underline{c}^T(n-1)\cdot\underline{x}(n)\right)\cdot\underline{x}(n)$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu\alpha\,\underline{c}(n-1) + \mu\left(d(n) - \underline{c}^T(n-1)\cdot\underline{x}(n)\right)\cdot\underline{x}(n)$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu\alpha\,\underline{c}(n-1) + \mu\, d(n)\cdot\underline{x}(n) - \mu\,\underbrace{\left(\underline{c}^T(n-1)\cdot\underline{x}(n)\right)}_{\text{skalar!}}\cdot\underline{x}(n)$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu\cdot\alpha\cdot\underline{c}(n-1) + \mu\, d(n)\,\underline{x}(n) - \mu\cdot\underline{x}(n)\cdot\underline{c}^T(n-1)\cdot\underline{x}(n)$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu\cdot\alpha\cdot\underline{c}(n-1) + \mu\,\underbrace{\left(\underline{h}^T\underline{x}(n)\right)}_{\text{scalar!}}\cdot\underline{x}(n) + \mu\cdot w(n)\,\underline{x}(n) - \mu\,\underline{x}(n)\cdot\underline{c}^T(n-1)\cdot\underline{x}(n)$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu\cdot\alpha\cdot\underline{c}(n-1) + \mu\,\underline{x}(n)\cdot\underline{h}^T\underline{x}(n) + \mu\cdot w(n)\cdot\underline{x}(n) - \mu\cdot\underline{x}(n)\cdot\underline{c}^T(n-1)\cdot\underline{x}(n)$$

$$\underline{c}(n) = \underline{c}(n-1) - \mu\alpha\,\underline{c}(n-1) + \mu\cdot\underline{x}(n)\cdot\underline{x}^T(n)\cdot\underline{h} + \mu\, w(n)\,\underline{x}(n) - \mu\,\underline{x}(n)\cdot\underline{x}^T(n)\cdot\underline{c}(n-1)$$

$$\cancel{E(\underline{c}(n))} = \cancel{E(\underline{c}(n-1))} - \mu\cdot\alpha\cdot E(\underline{c}(n-1)) + \mu\cdot \underbrace{E(\underline{x}(n)\cdot\underline{x}^T(n))}_{R_{xx}}\cdot\underline{h} + \mu\cdot E(w(n)\cdot\underline{x}(n)) - \mu\cdot \underbrace{E(\underline{x}(n)\cdot\underline{x}^T(n))}_{R_{xx}}\cdot\underline{c}(n-1))$$

$$\mu\alpha\, E(\underline{c}(n-1)) = \mu\cdot R_{xx}\cdot\underline{h} + \mu\cdot E(w(n)\,\underline{x}(n)) - \mu\cdot\underbrace{E(\underline{x}(n)\cdot\underline{x}^T(n))}_{R_{xx}}\cdot E(\underline{c}(n-1))$$

$$\cancel{\mu}\alpha\, E(\underline{c}(n-1)) = \cancel{\mu}\cdot R_{xx}\cdot\underline{h} + \cancel{\mu}\cdot\underbrace{E(w(n)\cdot\underline{x}(n))}_{\underline{p}} - \cancel{\mu}\cdot R_{xx}\cdot E(\underline{c}(n-1))$$

$$\alpha\, E(\underline{c}(n-1)) = R_{xx}\cdot\underline{h} + \underline{p} - R_{xx}\, E(\underline{c}(n-1))$$

$$\alpha\, E(\underline{c}(n-1)) + R_{xx}\, E(\underline{c}(n-1)) = R_{xx}\cdot\underline{h} + \underline{p}$$

$$(\alpha\cdot\underline{I} + R_{xx})\cdot E(\underline{c}(n-1)) = R_{xx}\cdot\underline{h} + \underline{p}$$

$$E(\underline{c}(n-1)) = (\alpha\cdot\underline{I} + R_{xx})^{-1}\cdot(R_{xx}\cdot\underline{h} + \underline{p})$$

$$\underline{E(\underline{c}_\infty) = (\alpha\cdot\underline{I} + R_{xx})^{-1}\cdot(R_{xx}\cdot\underline{h} + \underline{p})}$$

white Gaussian noise $x(n): \sigma_x^2$
white Gaussian noise $w(n): \sigma_w^2$
} uncorrelated $E(\underline{x}(n)\cdot w(n)) = E(w(n))\cdot E(\underline{x}(n))$

$$E(w(n)) = 0 \;\rightarrow\; \underline{p} = E(w(n)\cdot\underline{x}(n)) = E(w(n))\cdot E(\underline{x}(n)) = 0\cdot E(\underline{x}(n)) = \underline{0}$$

$$E(\underline{c}_\infty) = (\alpha\cdot\underline{I} + R_{xx})^{-1}\cdot R_{xx}\cdot\underline{h}$$

Observation:

$$R_{xx} = \begin{pmatrix}\sigma_x^2 & & 0\\ & \ddots & \\ 0 & & \sigma_x^2\end{pmatrix} \text{ ... white Gaussian noise}$$

$$\begin{pmatrix}\alpha+\sigma_x^2 & & \\ & \ddots & \\ & & \alpha+\sigma_x^2\end{pmatrix}^{-1}\begin{pmatrix}\sigma_x^2 & & 0\\ & \ddots & \\ 0 & & \sigma_x^2\end{pmatrix}\cdot\underline{h} = \begin{pmatrix}\frac{1}{\alpha+\sigma_x^2} & & \\ & \ddots & \\ & & \frac{1}{\alpha+\sigma_x^2}\end{pmatrix}\cdot\begin{pmatrix}\sigma_x^2 & & 0\\ & \ddots & \\ 0 & & \sigma_x^2\end{pmatrix}\cdot\underline{h}$$

$$\begin{pmatrix}\frac{\sigma_x^2}{\alpha+\sigma_x^2} & & 0\\ & \ddots & \\ 0 & & \frac{\sigma_x^2}{\alpha+\sigma_x^2}\end{pmatrix}\cdot\underline{h}$$

$\rightarrow$ each h_i is scaled by $\frac{\sigma_x^2}{\alpha+\sigma_x^2}$ (<1), e.g. $h_1\cdot 0{,}55$

$\rightarrow$ $E(c_\infty)$ will be never $\underline{h}$

Assignment 2

2.3.b) $\alpha > 0$: Avoid division by 0 if $\|\underline{x}(n)\|^2 = 0$, because this may happen sometimes

$\rightarrow$ improve stability

$\|\underline{x}(n)\|^2$: normalizes input power

$\|\underline{x}(n)\|^2 >> \rightarrow \frac{\mu}{\|\underline{x}(n)\|^2 + \alpha} <<$ but $\frac{\underline{x}(n)}{\|\underline{x}(n)\|^2}$ remains „constant" $\rightarrow$

avoids big fluctuation in update step

$\|\underline{x}(n)\|^2 << \rightarrow \frac{\mu}{\|\underline{x}(n)\|^2 + \alpha} >>$ but $\frac{\underline{x}(n)}{\|\underline{x}(n)\|^2}$ remains „constant" $\rightarrow$

avoids big fluctuations in update step

$\rightarrow$ constant convergence steps/speed

Assignment 2, 2.6a)

$$J_{MSE}(\underline{c}) = E(e(n)\cdot e^*(n))$$

$$\nabla_{\underline{c}}(J_{MSE}) = \nabla_{\underline{c}_R}(J_{MSE}) + j\nabla_{\underline{c}_i}(J_{MSE}) = \nabla_{\underline{c}_R}(E(e(n)\cdot e^*(n))) + j\nabla_{\underline{c}_i}(E(e(n)\cdot e^*(n))) = 0$$

$$\nabla_{\underline{c}_R}(E(e(n)\cdot e^*(n))) = E(\underbrace{\nabla_{\underline{c}_R}(e(n)\cdot e^*(n))}_{I}) + jE(\underbrace{\nabla_{\underline{c}_i}(e(n)\cdot e^*(n))}_{II}) = 0$$

$$I \quad \nabla_{\underline{c}_R}(e(n)\cdot e^*(n)) = \underbrace{\nabla_{\underline{c}_R}(e(n))}_{III}\cdot e^*(n) + e(n)\cdot \underbrace{\nabla_{\underline{c}_R}(e^*(n))}_{IV}$$

$$II \quad \nabla_{\underline{c}_i}(e(n)\cdot e^*(n)) = \underbrace{\nabla_{\underline{c}_i}(e(n))}_{V}\cdot e^*(n) + e(n)\cdot \underbrace{\nabla_{\underline{c}_i}(e^*(n))}_{VI}$$

$$III \quad \nabla_{\underline{c}_R}(e(n)) = \nabla_{\underline{c}_R}(y(n)-d(n)) = \nabla_{\underline{c}_R}(\underline{c}^H\cdot\underline{x}(n) - \underbrace{d(n)}_{0}) = \nabla_{\underline{c}_R}((\underline{c}_R + j\underline{c}_i)^H\cdot\underline{x}(n)) = \nabla_{\underline{c}_R}(\underline{c}_R^H\cdot\underline{x}(n) + j\underline{c}_i^H\cdot\underline{x}(n)) =$$

$$\nabla_{\underline{c}_R}(\underline{c}_R^H\cdot\underline{x}(n)) + \underbrace{\nabla_{\underline{c}_R}(j\underline{c}_i^H\underline{x}(n))}_{0} = \nabla_{\underline{c}_R}(\underline{c}_R^H\underline{x}(n)) = \underline{x}(n)$$

$$IV \quad \nabla_{\underline{c}_R}(e^*(n)) = \nabla_{\underline{c}_R}((y(n)-d(n))^*) = \nabla_{\underline{c}_R}(y^*(n) - d^*(n)) = \nabla_{\underline{c}_R}(\underline{c}^{H*}\cdot\underline{x}^*(n) - d^*(n)) = \nabla_{\underline{c}_R}(\underline{c}^T\cdot\underline{x}^*(n) - d^*(n)) =$$

$$\nabla_{\underline{c}_R}((\underline{c}_R + j\underline{c}_i)^T\underline{x}^*(n) - d^*(n)) = \nabla_{\underline{c}_R}(\underline{c}_R^T\underline{x}^*(n) + \underbrace{j\underline{c}_i^T\underline{x}^*(n)}_{0} - \underbrace{d^*(n)}_{0}) = \underline{x}^*(n)$$

$$V \quad \nabla_{\underline{c}_i}(e(n)) = \nabla_{\underline{c}_i}(y(n)-d(n)) = \nabla_{\underline{c}_i}(\underline{c}^H\underline{x}(n) - d(n)) = \nabla_{\underline{c}_i}((\underline{c}_R + j\underline{c}_i)^H\cdot\underline{x}(n) - d(n)) =$$

$$\nabla_{\underline{c}_i}((\underline{c}_R + j\underline{c}_i)^{*T}\cdot\underline{x}(n) - d(n)) = \nabla_{\underline{c}_i}((\underline{c}_R - j\underline{c}_i)^T\cdot\underline{x}(n) - d(n)) = \nabla_{\underline{c}_i}(\underbrace{\underline{c}_R^T\underline{x}(n)}_{0} - j\underline{c}_i^T\underline{x}(n) - \underbrace{d(n)}_{0}) = -j\underline{x}(n)$$

$$VI \quad \nabla_{\underline{c}_i}(e^*(n)) = \nabla_{\underline{c}_i}((y(n)-d(n))^*) = \nabla_{\underline{c}_i}((\underline{c}^H\underline{x}(n) - d(n))^*) = \nabla_{\underline{c}_i}(\underline{c}^{H*}\underline{x}^*(n) - d^*(n)) = \nabla_{\underline{c}_i}(\underline{c}^T\underline{x}^*(n) - d^*(n)) =$$

$$\nabla_{\underline{c}_i}((\underline{c}_R + j\underline{c}_i)^T\underline{x}^*(n) - d^*(n)) = \nabla_{\underline{c}_i}(\underbrace{\underline{c}_R^T\underline{x}^*(n)}_{0} + j\underline{c}_i^T\underline{x}^*(n) - \underbrace{d^*(n)}_{0}) = j\underline{x}^*(n)$$

$$\nabla_{\underline{c}}(J_{MSE}) = \nabla_{\underline{c}_R}(J_{MSE}) + j\nabla_{\underline{c}_i}(J_{MSE}) = \nabla_{\underline{c}_R}(E(e(n)\cdot e^*(n))) + j\nabla_{\underline{c}_i}(E(e(n)\cdot e^*(n))) = 0$$

$$E(\underbrace{\nabla_{\underline{c}_R}(e(n)\cdot e^*(n))}_{I}) + jE(\underbrace{\nabla_{\underline{c}_i}(e(n)\cdot e^*(n))}_{II}) = 0$$

$$E(e^*(n)\cdot III + e(n)\cdot IV) + j\cdot E(e^*(n)\cdot V + e(n)\cdot VI) = 0$$

$$E(e^*(n)\cdot\underline{x}(n) + e(n)\cdot\underline{x}^*(n)) + jE(e^*(n)\cdot(-j\underline{x}(n)) + e(n)\,j\underline{x}^*(n)) = 0$$

$$E(e^*(n)\cdot\underline{x}(n) + e(n)\cdot\underline{x}^*(n) - j^2 e^*(n)\underline{x}(n) + j^2 e(n)\cdot\underline{x}^*(n)) = 0$$

$$E(e^*(n)\cdot\underline{x}(n) + \cancel{e(n)\cdot\underline{x}^*(n)} + e^*(n)\cdot\underline{x}(n) - \cancel{e(n)\cdot\underline{x}^*(n)}) = 0$$

$$2E(e^*(n)\cdot\underline{x}(n)) = 0 \qquad | \; e(n) = \underline{c}^H\underline{x}(n) - \underline{h}^H\underline{x}(n) - w(n)$$

subst. e)

$$E((\underline{c}^H\underline{x}(n) - \underline{h}^H\underline{x}(n) - w(n))^*\cdot\underline{x}(n)) = E(\underline{c}^{H*}\underline{x}^*(n)\underline{x}(n) - \underline{h}^{H*}\underline{x}^*(n)\cdot\underline{x}(n) - w^*(n)\cdot\underline{x}(n)) =$$

$$E(\underline{c}^T\underline{x}^*(n)\cdot\underline{x}(n) - \underline{h}^T\underline{x}^*(n)\cdot\underline{x}(n) - w^*(n)\cdot\underline{x}(n)) = E(\underline{x}(n)\cdot\underline{c}^T\underline{x}^*(n) - \underline{x}(n)\cdot\underline{h}^T\underline{x}^*(n) - w^*(n)\cdot\underline{x}(n)) =$$

$$E(\underline{x}(n)\cdot\underline{x}^{*T}(n)\cdot\underline{c} - \underline{x}(n)\cdot\underline{x}^{*T}(n)\cdot\underline{h} - w^*(n)\cdot\underline{x}(n)) = \underbrace{E(\underline{x}(n)\cdot\underline{x}^H(n))}_{R_{xx}}\cdot\underline{c} - \underbrace{E(\underline{x}(n)\cdot\underline{x}^H(n))}_{R_{xx}}\cdot\underline{h} - \underbrace{E(w^*(n)\cdot\underline{x}(n))}_{\tilde{\underline{p}}} = 0$$

$$\underline{R}_{xx}\cdot\underline{c} - \underline{R}_{xx}\cdot\underline{h} - \tilde{\underline{p}} = 0 \quad \rightarrow \quad \underline{R}_{xx}\cdot\underline{c} = \underline{R}_{xx}\cdot\underline{h} + \tilde{\underline{p}}$$

$$\underline{\underline{c}_{MSE} = \underline{R}_{xx}^{-1}(\underline{R}_{xx}\cdot\underline{h} + \tilde{\underline{p}})} \qquad \tilde{\underline{p}} = 0 \;\rightarrow\; \underline{\underline{c}_{MSE} = \underline{h}}$$

Assignment 2, 2.6a)

subst. b) $2E(e^*(n)\cdot \underline{x}(n))=0$ $\quad | \; e(n)=\underline{c}^H \underline{x}(n)-d(n)$

$$E((\underline{c}^{H^*}\underline{x}^*(n)-d^*(n))\cdot\underline{x}(n))=0$$

$$E(\underline{c}^T\underline{x}^*(n)\cdot\underline{x}(n)-d^*(n)\,\underline{x}(n))=0$$

$$E(\underline{x}(n)\cdot\underline{c}^T\cdot\underline{x}^*(n)-d^*(n)\cdot\underline{x}(n))=0$$

$$E(\underline{x}(n)\cdot\underline{x}^{*T}(n)\cdot\underline{c}-d^*(n)\cdot\underline{x}(n))=0$$

$$\underbrace{E(\underline{x}(n)\cdot\underline{x}^H(n))}_{\underline{R}_{xx}}\cdot\underline{c}-\underbrace{E(d^*(n)\cdot\underline{x}(n))}_{\underline{p}}=0$$

$$\underline{R}_{xx}\cdot\underline{c}=\underline{p}$$

$$\underline{\underline{c}=\underline{R}_{xx}^{-1}\cdot\underline{p}}$$ (if only „$d(n)$" is known)

Assignment 2, 2.6.b)

$$J_{MSE}(\underline{c}) = E\left(e(n)\cdot e^{*}(n)\right) = E\left(\left(\underline{c}^{H}\underline{x}(n) - \underline{h}^{H}\underline{x}(n) - w(n)\right)\cdot\left(\underline{c}^{H^{*}}\underline{x}^{*}(n) - \underline{h}^{H^{*}}\cdot \underline{x}^{*}(n) - w^{*}(n)\right)\right) =$$

$$E\Big(\underline{c}^{H}\underline{x}(n)\cdot \underline{c}^{T}\underline{x}^{*}(n) - \underline{c}^{H}\underline{x}(n)\,\underline{h}^{T}\underline{x}^{*}(n) - \underbrace{\underline{c}^{H}\underline{x}(n)\,w^{*}(n)}_{\text{independent}} - \underline{h}^{H}\underline{x}(n)\cdot\underline{c}^{T}\underline{x}^{*}(n) + \underline{h}^{H}\underline{x}(n)\,\underline{h}^{T}\underline{x}^{*}(n) + \underbrace{\underline{h}^{H}\underline{x}(n)\,w^{*}(n)}_{\text{independent}}$$

$$- \underbrace{w(n)\cdot\underline{c}^{T}\underline{x}^{H}(n)}_{\text{indep.}} + \underbrace{w(n)\cdot\underline{h}^{T}\underline{x}^{*}(n)}_{\text{indep.}} + w(n)\cdot w^{*}(n)\Big) =$$

$$E\left(\underline{c}^{H}\cdot\underline{x}(n)\cdot\underline{x}^{*^{T}}(n)\cdot\underline{c} - \underline{c}^{H}\underline{x}(n)\cdot\underline{x}^{*^{T}}(n)\cdot\underline{h} - \underline{h}^{H}\underline{x}(n)\cdot\underline{x}^{*^{T}}(n)\cdot\underline{c} + \underline{h}^{H}\underline{x}(n)\cdot\underline{x}^{*^{T}}(n)\cdot\underline{h} + w(n)\cdot w^{*}(n)\right) =$$

$$E\left(\underline{c}^{H}\cdot\underline{x}(n)\cdot\underline{x}^{H}(n)\cdot\underline{c}\right) - E\left(\underline{c}^{H}\cdot\underline{x}(n)\cdot\underline{x}^{H}(n)\cdot\underline{h}\right) - \underline{h}^{H}E\left(\underline{x}(n)\cdot\underline{x}^{H}(n)\right)\cdot\underline{c} + \underline{h}^{H}E\left(\underline{x}(n)\cdot\underline{x}^{H}(n)\right)\cdot\underline{h} + \sigma_w^2 =$$

$$\underline{\underline{c}^{H}\cdot\underline{R}_{xx}\cdot\underline{c} - \underline{c}^{H}\cdot\underline{R}_{xx}\cdot\underline{h} - \underline{h}^{H}\underline{R}_{xx}\cdot\underline{c} + \underline{h}^{H}\underline{R}_{xx}\cdot\underline{h} + \sigma_w^2 = J_{MSE}(\underline{c})}$$

$$J_{MSE}(\underline{c}_{MSE}) = \underline{h}^{H}\cancel{\underline{R}_{xx}\underline{h}} - \underline{h}^{H}\cancel{\underline{R}_{xx}\underline{h}} - \underline{h}^{H}\cancel{\underline{R}_{xx}\underline{h}} + \underline{h}^{H}\cancel{\underline{R}_{xx}\underline{h}} + \sigma_w^2 = \sigma_w^2$$

$$\underline{J_{MSE}(\underline{c}_{MSE}) = \sigma_w^2}$$