ML Detection with Blind Linear Prediction for Differential Space-Time Block Code Systems

Seree Wanichpakdeedecha, Kazuhiko Fukawa,
Hiroshi Suzuki, Satoshi Suyama

Tokyo Institute of Technology

Background

STBC (Space-Time Block Codes)

DSTBC (Differential Space-Time Block Codes)

Differential Detection

- **Merit:** Channel estimation is not required.
- **Demerit:** Degradation of BER under static condition (3 dB E_b/N_0)

Linear Prediction Detection

- Suppressing the above degradation
- Tracking fast fading

Conventional : need fading information or training

Proposal: *Blind Linear Prediction (BLP) Detection*

- Perform linear prediction without fading info. nor training

Linear Prediction:
Conventional vs. Proposal

Conventional → **Proposal**

Conventional

- **Yule-Walker Equation**

  - Require **information on $f_D T$** and **SNR**

- **RLS algorithm**

  - Adaptively update the coefficients

  - Might need a lot of **training symbols** for accurate parameter estimation

Proposal

- **Blind Linear Prediction (BLP)**

- Determine constant prediction coefficients by the method of Lagrange multipliers.

- Does **NOT** need knowledge of
  1. $f_D T$
  2. SNR
  3. Training sequences

Signal Model

k: STBC block index
l: Rx index, p: Tx index

$\mathbf{s}^{\mathrm{T}}(k) = \left(A(k) \quad B(k)\right)\mathbf{D}(k-1)$

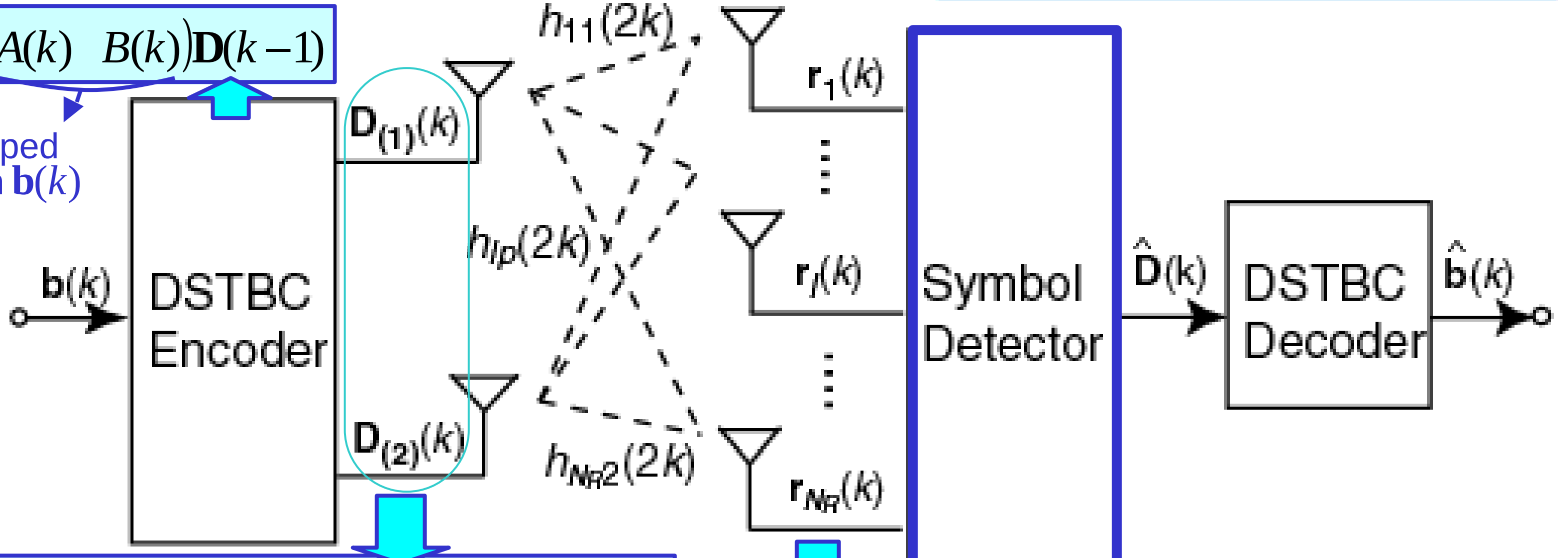
mapped from $\mathbf{b}(k)$

Assumption:

$h_{lp}(2k) \approx h_{lp}(2k-1)$

time-varying flat Rayleigh fading

$\mathbf{h}_l^H(k) = \left(h_{l1}^*(2k) \quad h_{l2}^*(2k)\right)$



Unitary matrix

$$\mathbf{D}(k) = \begin{pmatrix} s(2k-1) & s(2k) \\ -s^*(2k) & s^*(2k-1) \end{pmatrix}$$

with columns $\mathbf{D}_{(1)}(k)$ and $\mathbf{D}_{(2)}(k)$, and first row $\mathbf{s}^{\mathrm{T}}(k)$

$$\mathbf{r}_l(k) = \mathbf{D}(k)\mathbf{h}_l(k) + \mathbf{n}_l(k)$$

$$\mathbf{r}_l^H(k) = \left(r_l^*(2k-1) \quad r_l^*(2k)\right)$$

AWGN

ML Detection with Linear Prediction

■ Maximum Likelihood Detection:

Log likelihood function:

$$L = -\sum_{l=1}^{N_R} \sum_{k} \left\| \mathbf{D}_n^{\mathrm{H}}(k)\mathbf{r}_l(k) - \hat{\mathbf{h}}_l(k) \right\|^2$$

$\mathbf{D}_n(k)$: the n-th candidate of the unitary matrix $\mathbf{D}(k)$

$\hat{\mathbf{h}}_l(k)$: the estimate of $\mathbf{h}_l(k)$

$$\mathbf{D}^{\mathrm{H}}(k)\mathbf{r}_l(k) \quad = \quad \mathbf{h}_l(k) + \mathbf{D}^{\mathrm{H}}(k)\mathbf{n}_l(k)$$

autocorrelation = autocorrelation of $\mathbf{n}_l(k) = 2s_n^2$

■ Linear Prediction:

$\hat{\mathbf{h}}_l(k)$ is approximated as

$$\hat{\mathbf{h}}_l(k) = \sum_{m=1}^{M} c_m \mathbf{D}_n^{\mathrm{H}}(k-m)\mathbf{r}_l(k-m)$$

M : order of the prediction

$\{c_m\}$: prediction coefficients

Blind criteria to determine $\{c_m\}$

Log-Likelihood function

$$L = -\sum_{l=1}^{N_R} \sum_{k} \left\| \mathbf{e}_l(k) \right\|^2$$

$$\mathbf{e}_l(k) = \mathbf{D}^{\mathrm{H}}(k)\mathbf{r}_l(k) - \sum_{m=1}^{M} c_m \mathbf{D}^{\mathrm{H}}(k-m)\mathbf{r}_l(k-m)$$

Criteria: minimize $\left\langle \left\| \mathbf{e}_l(k) \right\|^2 \right\rangle$

$$\mathbf{e}_l(k) = \left[\mathbf{D}^{\mathrm{H}}(k)\mathbf{n}_l(k) - \sum_{m=1}^{M} c_m \mathbf{D}^{\mathrm{H}}(k)\mathbf{n}_l(k-m) \right] + \left[\mathbf{h}_l(k) - \sum_{m=1}^{M} c_m \mathbf{h}_l(k-m) \right]$$

$\text{ß}_l(k)$

= *zero*

(a constraint of h_l, AR process)

Noise minimization

min $\left\langle \left\| \text{ß}_l(k) \right\|^2 \right\rangle = 2s_n^2 \left(1 + \sum_{m=1}^{M} \left| c_m \right|^2 \right)$ → ***min*** $\sum_{m=1}^{M} \left| c_m \right|^2 = \mathbf{c}^{\mathrm{H}}\mathbf{c}$

$$\mathbf{c}^{\mathrm{H}} = \left(c_1, c_2, \text{☹}, c_M \right)$$

Modified constraints of $\{c_m\}$

From the channel constraints,

$$\mathbf{h}_l(k) = \sum_{m=1}^{M} c_m \mathbf{h}_l(k-m)$$

Taylor series expansion

$$\mathbf{h}_l(k-m) = \sum_{q=0}^{q_1} \frac{(-2m)^q}{q!} \mathbf{h}_l^{(q)}(k)$$

q_1: degree of the polynomial approximation

Modified Constraints:

$$\mathbf{c}^{\mathrm{H}} \mathbf{b}_0 = 1$$

$$\mathbf{c}^{\mathrm{H}} \mathbf{b}_q = 0, \quad 1 \le q \le q_1$$

$$\mathbf{b}_0^{\mathrm{H}} = (1,\ 1,\ ☺,\ 1);$$

$$\mathbf{b}_q^{\mathrm{H}} = ((-2)^q \quad (-4)^q \quad ☹ \quad (-2M)^q)$$

The Solutions of $\{c_m\}$

- ***The minimization of*** $\mathbf{c}^{\mathrm{H}}\mathbf{c}$

  under *the constraints*
  $$\mathbf{c}^{\mathrm{H}}\mathbf{b}_0 = 1$$
  $$\mathbf{c}^{\mathrm{H}}\mathbf{b}_q = 0, \quad 1 \le q \le q_1$$

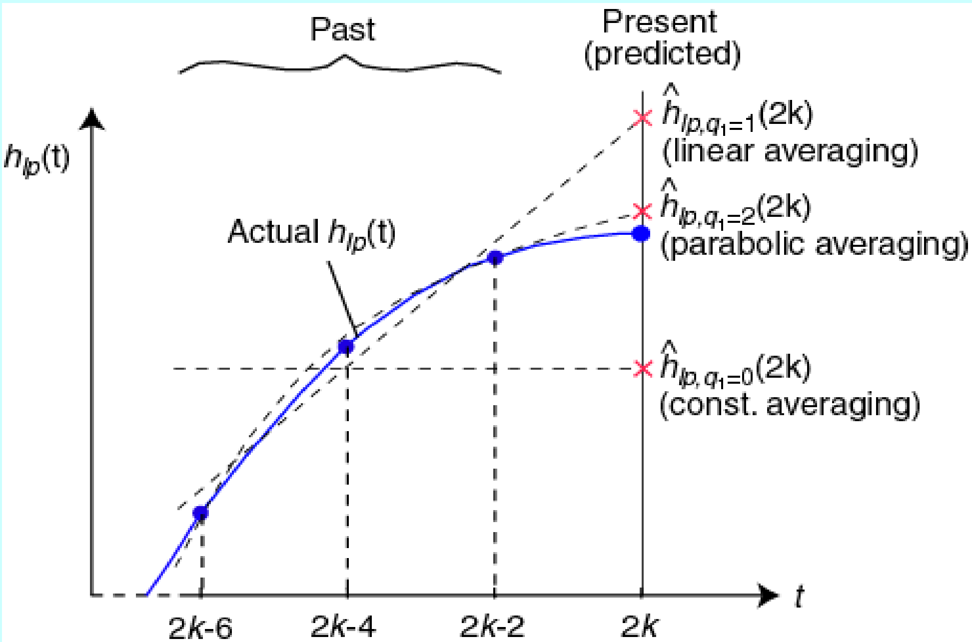
can be solved by using the method of Lagrange multipliers.

The solution of the coefficients:

$$\mathbf{c} = \sum_{q=0}^{q_1} (\mathbf{B}_{q_1}^{-1})_{q+1,1}\mathbf{b}_q$$

$$(\mathbf{B})_{ij} = \mathbf{b}_{i-1}^{H}\mathbf{b}_{j-1}, \quad 1 \le i, j \le q_1 + 1$$

Solutions Analysis



The solutions:

$$\mathbf{C} = \sum_{q=0}^{q_1} (\mathbf{B}_{q_1}{}^{-1})_{q+1,1} \mathbf{b}_q$$

$M = 3$:

- $q_1 = 0$ → Constant ave.
- $q_1 = 1$ → Linear ave.
- $q_1 = 2$ → Parabolic ave.

Larger q_1 ➡ **Prediction of h_{lp} becomes more accurate**

trade-off

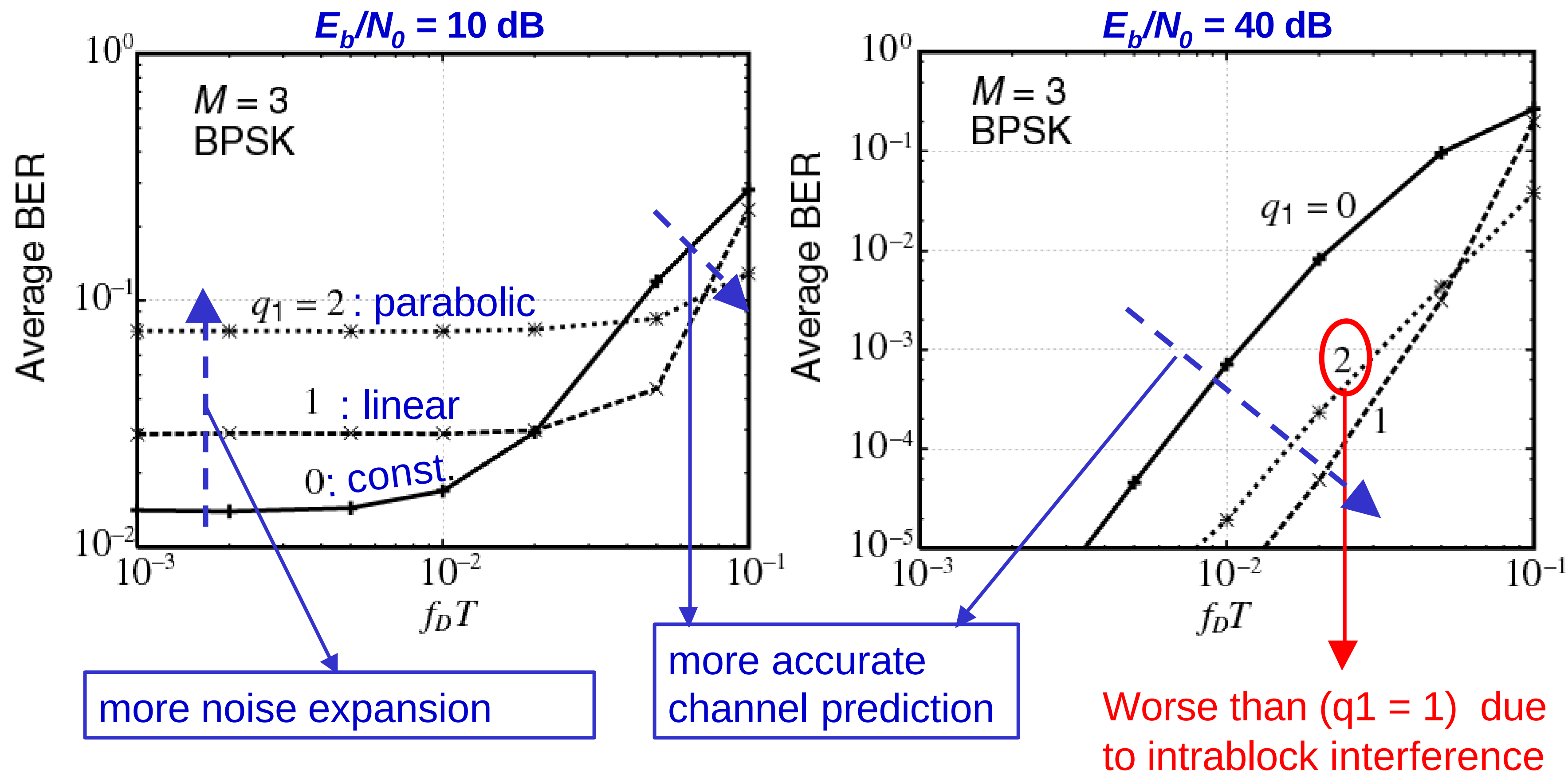
However..

Larger q_1 ➡ **$\{c_m\}$ has larger variance** (due to increasing constraints) ➡ Larger $\sum_{m=1}^{M} |c_m|^2$ ➡ **More noise expansion**

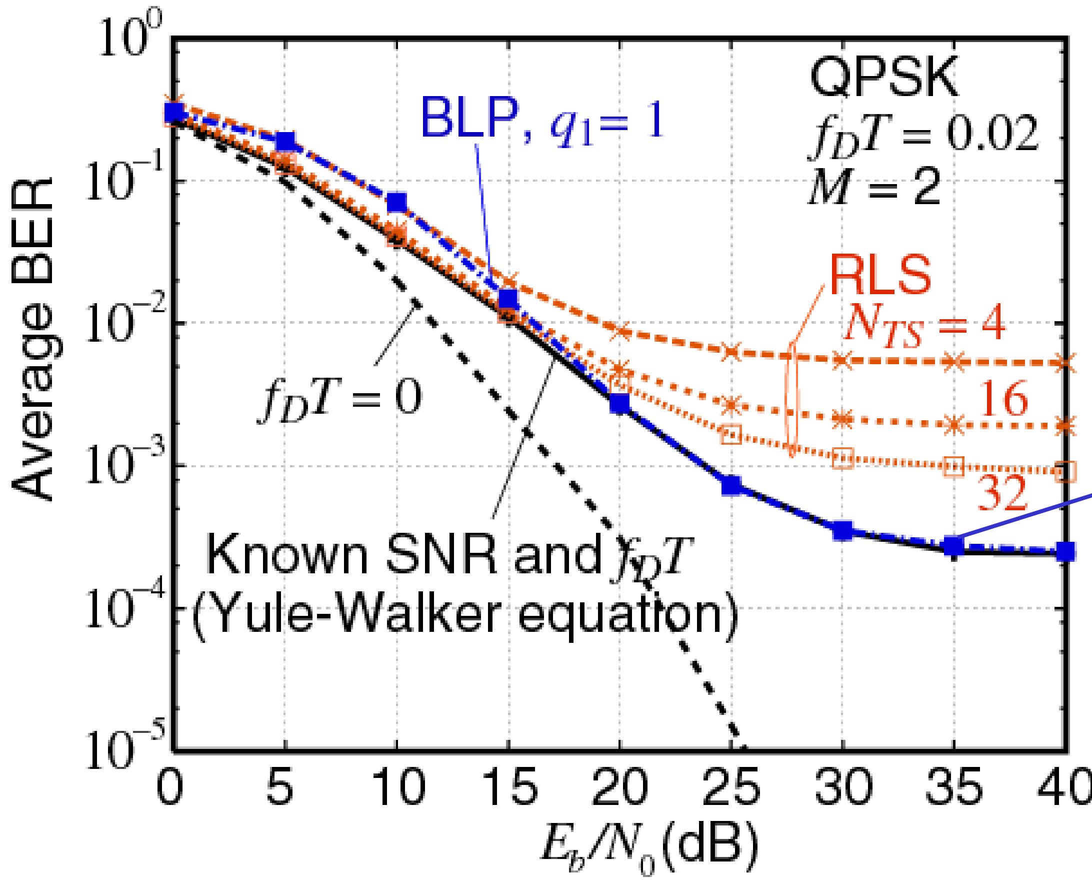
Simulation Conditions

Number of symbol per packet	128
Constellation mapping	BPSK, QPSK
Number of transmit antenna	2
Number of receive antenna	1
Channel	Fast flat Rayleigh fading
Normalized Doppler frequency ($f_D T$)	1×10^{-3} – 1×10^{-1}

Dependence on Degree of Polynomial q_1



Average BER Performance vs. E_b/N_0



N_{TS} = Number of training symbols (per 128 data symbols)

BLP is Better than RLS (32 N_{TS}) in both BER performance and transmission efficiency

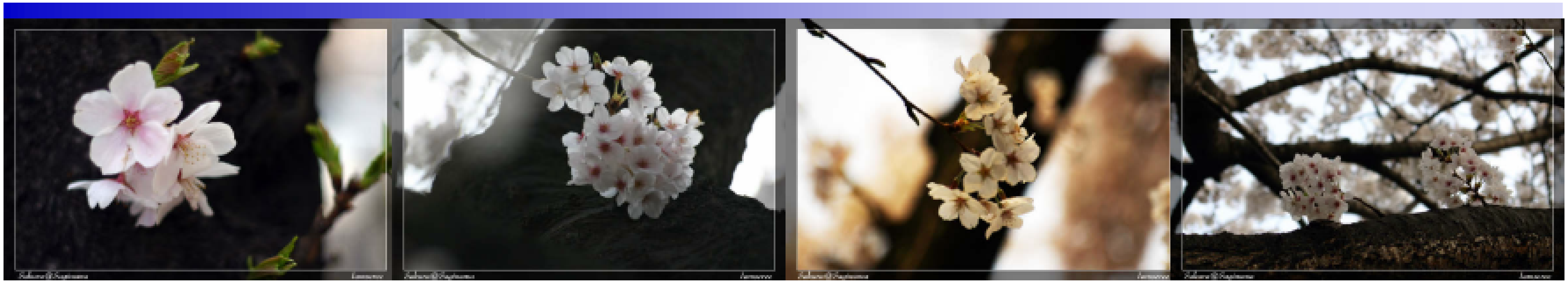
Conclusion

- **Blind Linear Prediction (BLP) on Differential STBC (DSTBC)** has been proposed by the method of Lagrange multipliers.
  - **NO** ***channel information*** nor ***training sequences***

- The BER performance of BLP is **better** than that of the RLS algorithm with 32-symbol training.

- The performance of BLP depends on $\underline{\boldsymbol{q}}_1$ which is obtained by trade-off between
  - Accuracy of channel prediction.
  - Noise expansion.

Thank you!!



Solving $\{c_m\}$ by Yule-Walker Equation

Yule-Walker Equation

$$\begin{pmatrix} R_a(0)+g^{-1} & R_a(-2) & \cdots & R_a(-2M) \\ R_a(2) & R_a(0)+g^{-1} & \cdots & R_a[-2(M-1)] \\ \vdots & \vdots & \ddots & \vdots \\ R_a(2M) & R_a[2(M-1)] & \cdots & R_a(0)+g^{-1} \end{pmatrix} \begin{pmatrix} c_0 \\ -c_1 c_0 \\ \vdots \\ -c_M c_0 \end{pmatrix} = \begin{pmatrix} s_e^2 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

where

$R_a(i) = J_0(2pf_D T i)$: an auto-correlation function of $h_{lp}(2k)$

$g = \dfrac{\left\langle \left| h_{lp}(2k) \right|^2 \right\rangle}{s_n^2}$: SNR

$s_e^2 = \text{const.}$: the squared prediction error

Solving $\{c_m\}$ by RLS Algorithm

- Recursively, the prediction coefficients $\{c_m\}$ are updated using N_{TS} training symbols.

- **Linear Prediction Error**: $$\mathbf{?}(k) = \mathbf{z}(k) - \mathbf{Z}^{\mathrm{T}}(k)\mathbf{c}^{*}(k-1)$$
- **Gain matrix**: $$\mathbf{K}(k) = l^{-1}\mathbf{P}(k-1)\mathbf{Z}(k)(\mathbf{I} + l^{-1}\mathbf{Z}^{\mathrm{H}}(k)\mathbf{P}(k-1)\mathbf{Z}(k))^{-1}$$
- **Inverse of covariance matrix**: $$\mathbf{P}(k) = l^{-1}\mathbf{P}(k-1) - l^{-1}\mathbf{K}(k)\mathbf{Z}^{\mathrm{H}}(k)\mathbf{P}(k-1)$$
- **Coefficients Update**: $$\mathbf{c}(k) = \mathbf{c}(k-1) + \mathbf{K}(k)\mathbf{?}^{*}(k)$$

$$\mathbf{?}^{\mathrm{H}}(k) = \left(\mathbf{e}_1^{\mathrm{H}} \quad \mathbf{e}_2^{\mathrm{H}} \quad \text{☹} \quad \mathbf{e}_{N_R}^{\mathrm{H}}\right)$$

$$\mathbf{z}^{\mathrm{H}}(k) = \left(\mathbf{r}_1^{\mathrm{H}}(k)\mathbf{D}(k) \quad \mathbf{r}_2^{\mathrm{H}}(k)\mathbf{D}(k) \quad \text{☹} \quad \mathbf{r}_{N_R}^{\mathrm{H}}(k)\mathbf{D}(k)\right)$$

$$\mathbf{Z}^{\mathrm{H}}(k) = \left(\mathbf{z}^{*}(k-1) \quad \mathbf{z}^{*}(k-2) \quad \text{☹} \quad \mathbf{z}^{*}(k-M)\right)$$