Reduced Complexity Tracking of Doubly-Selective Channels

Arun P. Kannu and Phil Schniter

THE OHIO STATE UNIVERSITY

- Problem:

  Estimation and tracking of doubly-selective channels.

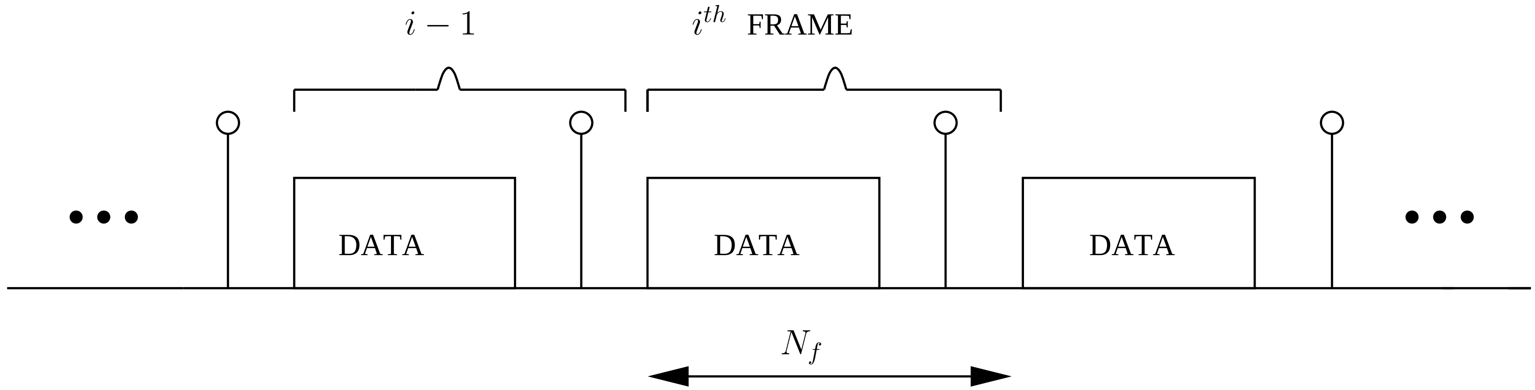
- Approach:

  Use current and previous pilots as well as previously decoded data.

- Goal:

  Near optimal performance without large matrix inversions.

- Transmission Model:



- Reception Model:

$$
\begin{aligned}
y_n^{(i)} &= \sum_{d=0}^{N_h-1} h_{n,d}^{(i)} t_{n-d}^{(i)} + v_n^{(i)} \\
\boldsymbol{y}_{\mathsf{d}}^{(i)} &= \boldsymbol{T}_{\mathsf{d}}^{(i)} \boldsymbol{h}_{\mathsf{d}}^{(i)} + \boldsymbol{v}_{\mathsf{d}}^{(i)} \\
\boldsymbol{y}_{\mathsf{p}}^{(i)} &= \boldsymbol{T}_{\mathsf{p}} \boldsymbol{h}_{\mathsf{p}}^{(i)} + \boldsymbol{v}_{\mathsf{p}}^{(i)}
\end{aligned}
$$

Reception Model (cont.):

$$\boldsymbol{y}^{(i)} = \begin{bmatrix} \boldsymbol{y}_{\mathsf{d}}^{(i)} \\ \boldsymbol{y}_{\mathsf{p}}^{(i)} \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{T}_{\mathsf{d}}^{(i)} & \\ & \boldsymbol{T}_{\mathsf{p}} \end{bmatrix}}_{\boldsymbol{T}^{(i)}} \underbrace{\begin{bmatrix} \boldsymbol{h}_{\mathsf{d}}^{(i)} \\ \boldsymbol{h}_{\mathsf{p}}^{(i)} \end{bmatrix}}_{\boldsymbol{h}^{(i)}} + \underbrace{\begin{bmatrix} \boldsymbol{v}_{\mathsf{d}}^{(i)} \\ \boldsymbol{v}_{\mathsf{p}}^{(i)} \end{bmatrix}}_{\boldsymbol{v}^{(i)}}$$

- Estimate $\boldsymbol{h}_{\mathsf{d}}^{(i)}$ given $\{\boldsymbol{y}^{(j)}\}_{j \le i}$ and $\{\boldsymbol{T}_{\mathsf{d}}^{(j)}\}_{j \le i-1}$ and $\boldsymbol{T}_{\mathsf{p}}$.
  (assume channel process is WSS)

Background:

- Pilot-aided Wiener Estimation (PW)
- Pilot-aided Decision-directed Wiener Estimation (PDW)
- Pilot-aided Decision-directed Kalman Estimation (PDK)

Pilot-aided Wiener Estimation:

- For M frames:

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_{\mathsf{p}}^{(i)} \\ \vdots \\ \boldsymbol{y}_{\mathsf{p}}^{(i-M)} \end{bmatrix}}_{\underline{\boldsymbol{y}}_{\mathsf{p}}^{(i)}} = \underbrace{\begin{bmatrix} \boldsymbol{T}_{\mathsf{p}} & & \\ & \ddots & \\ & & \boldsymbol{T}_{\mathsf{p}} \end{bmatrix}}_{\boldsymbol{T}_M} \underbrace{\begin{bmatrix} \boldsymbol{h}_{\mathsf{p}}^{(i)} \\ \vdots \\ \boldsymbol{h}_{\mathsf{p}}^{(i-M)} \end{bmatrix}}_{\underline{\boldsymbol{h}}_{\mathsf{p}}^{(i)}} + \begin{bmatrix} \boldsymbol{v}_{\mathsf{p}}^{(i)} \\ \vdots \\ \boldsymbol{v}_{\mathsf{p}}^{(i-M)} \end{bmatrix}$$

$$\hat{\boldsymbol{h}}_{\mathsf{d}}^{(i)}\Big|_{\mathsf{pilot}} = \underbrace{\boldsymbol{R}_{\underline{\boldsymbol{h}}_{\mathsf{p}},\boldsymbol{h}_{\mathsf{d}}}^{H} \boldsymbol{T}_M^H (\boldsymbol{T}_M \boldsymbol{R}_{\underline{\boldsymbol{h}}_{\mathsf{p}},\underline{\boldsymbol{h}}_{\mathsf{p}}} \boldsymbol{T}_M^H + \sigma_v^2 \boldsymbol{I})^{-1}}_{\text{LTI}} \underline{\boldsymbol{y}}_{\mathsf{p}}^{(i)}.$$

- Advantage: time-invariant estimator.
- Limitation: if $f_d > \frac{1}{2N_f}$, channel is undersampled.

Pilot-aided Decision-directed Wiener Estimation:

- For M frames:

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_{\mathsf{p}}^{(i)} \\ \boldsymbol{y}^{(i-1)} \\ \vdots \\ \boldsymbol{y}^{(i-M)} \end{bmatrix}}_{\underline{\boldsymbol{y}}_{\mathsf{w}}^{(i)}} = \underbrace{\begin{bmatrix} \boldsymbol{T}_{\mathsf{p}} & & & \\ & \boldsymbol{T}^{(i-1)} & & \\ & & \ddots & \\ & & & \boldsymbol{T}^{(i-M)} \end{bmatrix}}_{\boldsymbol{T}_{\mathsf{w}}^{(i)}} \underbrace{\begin{bmatrix} \boldsymbol{h}_{\mathsf{p}}^{(i)} \\ \boldsymbol{h}^{(i-1)} \\ \vdots \\ \boldsymbol{h}^{(i-M)} \end{bmatrix}}_{\underline{\boldsymbol{h}}_{\mathsf{w}}^{(i)}} + \begin{bmatrix} \boldsymbol{v}_{\mathsf{p}}^{(i)} \\ \boldsymbol{v}^{(i-1)} \\ \vdots \\ \boldsymbol{v}^{(i-M)} \end{bmatrix}$$

$$\hat{\boldsymbol{h}}_{\mathsf{d}}^{(i)}\Big|_{\mathsf{wiener}} = \boldsymbol{R}_{\underline{\boldsymbol{h}}_{\mathsf{w}},\boldsymbol{h}_{\mathsf{d}}}^{H} \boldsymbol{T}_{\mathsf{w}}^{(i)H} (\boldsymbol{T}_{\mathsf{w}}^{(i)} \boldsymbol{R}_{\underline{\boldsymbol{h}}_{\mathsf{w}},\underline{\boldsymbol{h}}_{\mathsf{w}}} \boldsymbol{T}_{\mathsf{w}}^{(i)H} + \sigma_v^2 \boldsymbol{I})^{-1} \underline{\boldsymbol{y}}_{\mathsf{w}}^{(i)}.$$

- Limitation: large matrix inversion per each frame.

Pilot-aided Decision-directed Kalman Estimation:

- AR dynamic model: $\boldsymbol{h}^{(i)} = \boldsymbol{A}_\mathsf{k}\boldsymbol{h}^{(i-1)} + \boldsymbol{D}_\mathsf{k}\boldsymbol{w}_\mathsf{k}^{(i-1)}$.
- Current observation: $\underline{\boldsymbol{y}}_\mathsf{k}^{(i-1)} = \begin{bmatrix} \boldsymbol{y}_\mathsf{p}^{(i)} \\ \boldsymbol{y}_\mathsf{d}^{(i-1)} \end{bmatrix}$.
- Together

$$\underline{\boldsymbol{y}}_\mathsf{k}^{(i-1)} = \boldsymbol{C}_\mathsf{k}^{(i-1)}\boldsymbol{h}^{(i-1)} + \underline{\boldsymbol{v}}_\mathsf{k}^{(i-1)},$$

where $\boldsymbol{C}_\mathsf{k}^{(i-1)}$ contains data, pilots and $\boldsymbol{A}_\mathsf{k}$,
and where $\boldsymbol{v}_\mathsf{k}^{(i-1)}$ contains noise and channel variation.

- Note: slightly non-standard due to $\boldsymbol{S}_\mathsf{k} = E\{\boldsymbol{w}_\mathsf{k}^{(i-1)}\boldsymbol{v}_\mathsf{k}^{(i)H}\} \neq 0$.

Kalman Estimation (cont.):

- MMSE optimal estimate of $\boldsymbol{h}^{(i)}$ using $\{\underline{\boldsymbol{y}}_{\mathsf{k}}^{(i-1)}, ..., \underline{\boldsymbol{y}}_{\mathsf{k}}^{(0)}\}$ is

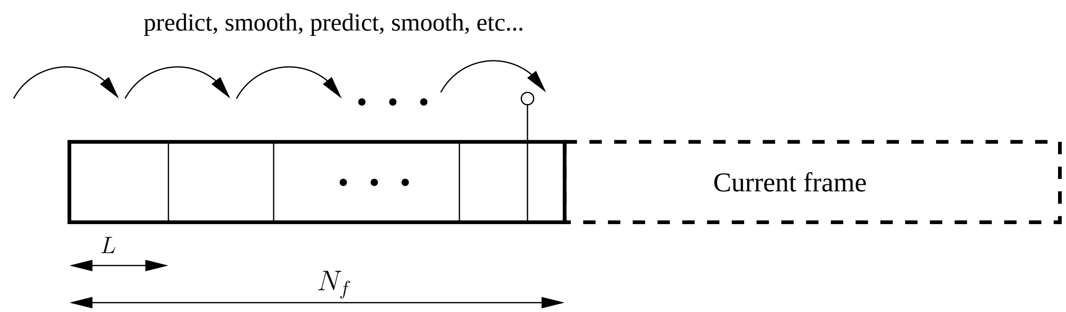
$$
\begin{aligned}
\hat{\boldsymbol{h}}^{(i)}\Big|_{\mathsf{kalman}} &= \boldsymbol{A}_{\mathsf{k}}\hat{\boldsymbol{h}}^{(i-1)}\Big|_{\mathsf{kalman}} + \boldsymbol{L}_{\mathsf{k}}^{(i-1)}\{\underline{\boldsymbol{y}}_{\mathsf{k}}^{(i-1)} - \boldsymbol{C}_{\mathsf{k}}^{(i-1)}\hat{\boldsymbol{h}}^{(i-1)}\Big|_{\mathsf{kalman}}\} \\
\boldsymbol{L}_{\mathsf{k}}^{(i-1)} &= (\boldsymbol{A}_{\mathsf{k}}\boldsymbol{P}_{\mathsf{k}}^{(i-1)}\boldsymbol{C}_{\mathsf{k}}^{(i-1)H} + \boldsymbol{D}_{\mathsf{k}}\boldsymbol{S}_{\mathsf{k}}) \\
&\quad \times (\boldsymbol{C}_{\mathsf{k}}^{(i-1)}\boldsymbol{P}_{\mathsf{k}}^{(i-1)}\boldsymbol{C}_{\mathsf{k}}^{(i-1)H} + \boldsymbol{R}_{\mathsf{k}})^{-1} \\
\boldsymbol{P}_{\mathsf{k}}^{(i-1)} &= \sigma_{w_{\mathsf{k}}}^2\boldsymbol{D}_{\mathsf{k}}\boldsymbol{D}_{\mathsf{k}}^H + \boldsymbol{A}_{\mathsf{k}}\boldsymbol{P}_{\mathsf{k}}^{(i-2)}\boldsymbol{A}_{\mathsf{k}}^H \\
&\quad - \boldsymbol{L}_{\mathsf{k}}^{(i-2)}(\boldsymbol{C}_{\mathsf{k}}^{(i-2)}\boldsymbol{P}_{\mathsf{k}}^{(i-2)}\boldsymbol{A}_{\mathsf{k}}^H + \boldsymbol{S}_{\mathsf{k}}^H\boldsymbol{D}_{\mathsf{k}}^H)
\end{aligned}
$$

where $\boldsymbol{R}_{\mathsf{k}} = E\{\underline{\boldsymbol{v}}_{\mathsf{k}}^{(i-1)}\underline{\boldsymbol{v}}_{\mathsf{k}}^{(i-1)H}\}$
with the initializations $\boldsymbol{P}_{\mathsf{k}}^{(0)} = E\{\boldsymbol{h}^{(0)}\boldsymbol{h}^{(0)H}\}$ and $\hat{\boldsymbol{h}}^{(0)}\Big|_{\mathsf{kalman}} = \mathbf{0}$.

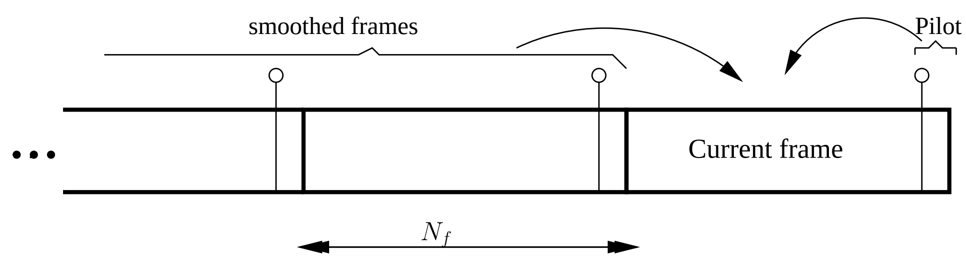
- Kalman estimation requires $N_f \times N_f$ matrix inversion per frame.

Low Complexity Predictor (LCP):

1. Compute Kalman smoothed channel in previous frame.



2. Wiener predict current channel using M past smoothed frames and current pilots.



Low Complexity Predictor (cont.):

Complexity of each stage (per frame):

1. Kalman Smoothing: $\frac{N_f}{L}$ matrix inversions of size $L \times L$.

   Note: L limits AR model size.

2. With some approximations, Wiener predictor is LTI.

   ($\Rightarrow$ no matrix inversion in prediction stage)

Approximations giving LTI Wiener Prediction:

$$\underbrace{\begin{bmatrix} \boldsymbol{y}_{\mathsf{p}}^{(i)} \\ \tilde{\boldsymbol{h}}^{(i-1)} \\ \vdots \\ \tilde{\boldsymbol{h}}^{(i-M)} \end{bmatrix}}_{\underline{\boldsymbol{y}}_{\mathsf{l}}^{(i)}} = \underbrace{\begin{bmatrix} \boldsymbol{T}_{\mathsf{p}} & & & \\ & \boldsymbol{I} & & \\ & & \ddots & \\ & & & \boldsymbol{I} \end{bmatrix}}_{\boldsymbol{B}} \underbrace{\begin{bmatrix} \boldsymbol{h}_{\mathsf{p}}^{(i)} \\ \boldsymbol{h}^{(i-1)} \\ \vdots \\ \boldsymbol{h}^{(i-M)} \end{bmatrix}}_{\underline{\boldsymbol{h}}_{\mathsf{l}}^{(i)}} + \begin{bmatrix} \boldsymbol{v}_{\mathsf{p}}^{(i)} \\ \boldsymbol{e}^{(i-1)} \\ \vdots \\ \boldsymbol{e}^{(i-M)} \end{bmatrix}$$

Assumptions on smoothing error $\boldsymbol{e}^{(j)}$:

1. White smoothing error (across time and lag)

$$E\{\boldsymbol{e}^{(j)}\boldsymbol{e}^{(k)H}\} = \sigma_v^2 \boldsymbol{I}\delta(j-k).$$

2. Smoothing errors uncorrelated with channel (across time and lag)

$$E\{\boldsymbol{h}^{(j)}\boldsymbol{e}^{(k)H}\} = \boldsymbol{0}.$$

$\Rightarrow$ Prediction: $\hat{\boldsymbol{h}}_{\mathsf{d}}^{(i)}\Big|_{\mathsf{lcp}} = \underbrace{\boldsymbol{R}_{\underline{\boldsymbol{h}}_{\mathsf{l}},\boldsymbol{h}_{\mathsf{d}}}^{H} \boldsymbol{B}^H (\boldsymbol{B}\boldsymbol{R}_{\underline{\boldsymbol{h}}_{\mathsf{l}},\underline{\boldsymbol{h}}_{\mathsf{l}}}\boldsymbol{B}^H + \sigma_v^2 \boldsymbol{I})^{-1}}_{\text{LTI}} \underline{\boldsymbol{y}}_{\mathsf{l}}^{(i)}.$

Modifications of LCP:

1. LCP with Kalman prediction (LCKP):
   - LTI M-frame Wiener Prediction replaced by steady-state Kalman prediction.
   - Uses only the previous smoothed frame.
2. LCP with Doppler-lag coefficients (LCPD):
   - Transform previous time-lag estimates to Doppler-lag representation.
   - Keep only significant Doppler indices for prediction of the current time-lag channel.
   - Factor of (almost) $\frac{1}{2f_d}$ savings !!

$\rightsquigarrow$ Reduce memory and computation.

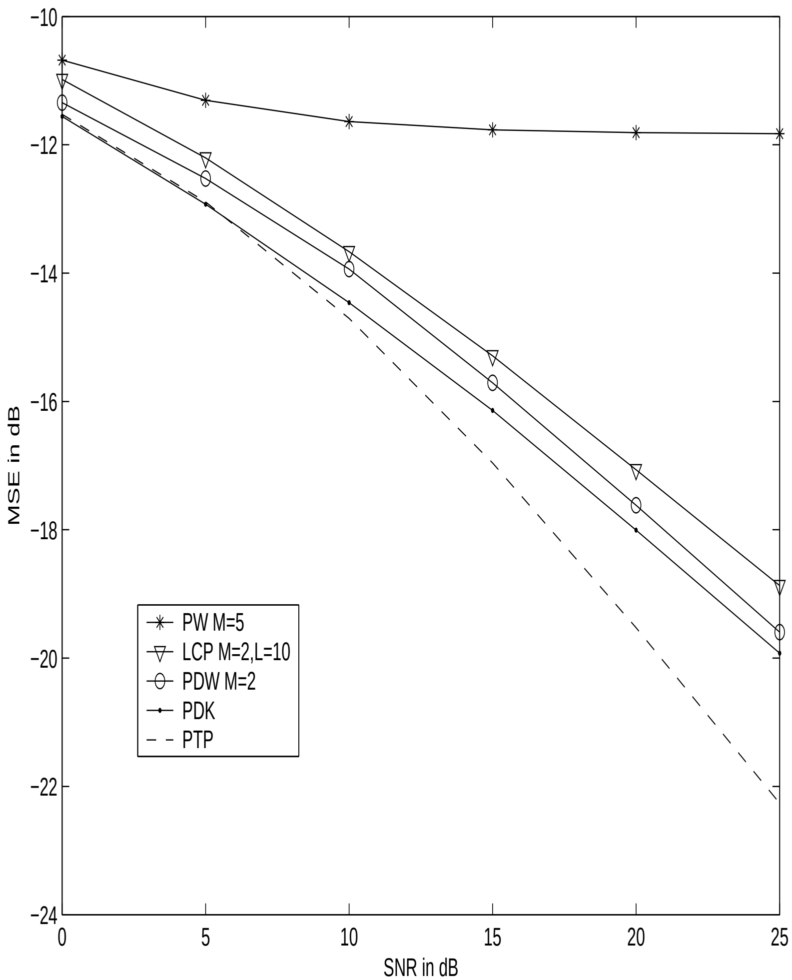
Simulation Results:

- Simulation parameters $N_f = 80$, $N_h = 8$.
- Relative algorithm complexity:

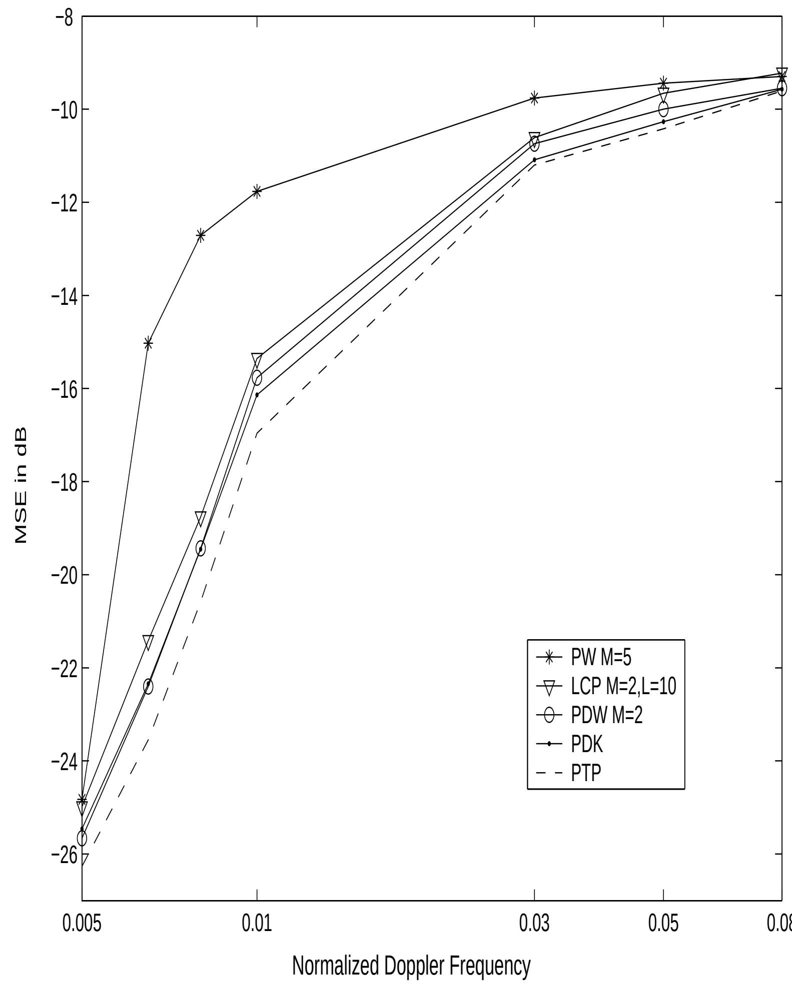
Technique	*Size of matrix inversion per frame*
PW	n.a.
PDW	$(MN_f + N_h) \times (MN_f + N_h)$
PDK	$N_f \times N_f$
LCP, LCPD and LCKP	$L \times L$

- Typical values are $L = 10$ and $M = 2$.
- Benchmark: IIR Wiener prediction with persistent Kronecker-delta pilot transmission (PTP).

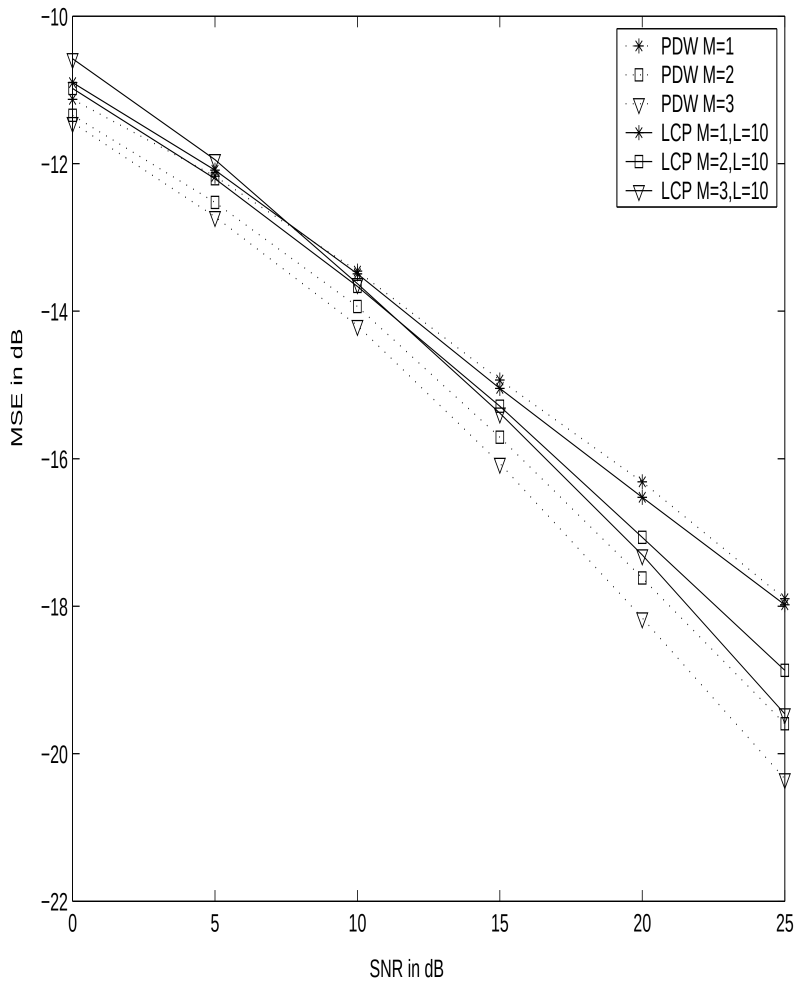
Effect of SNR @ $f_d = 0.01$



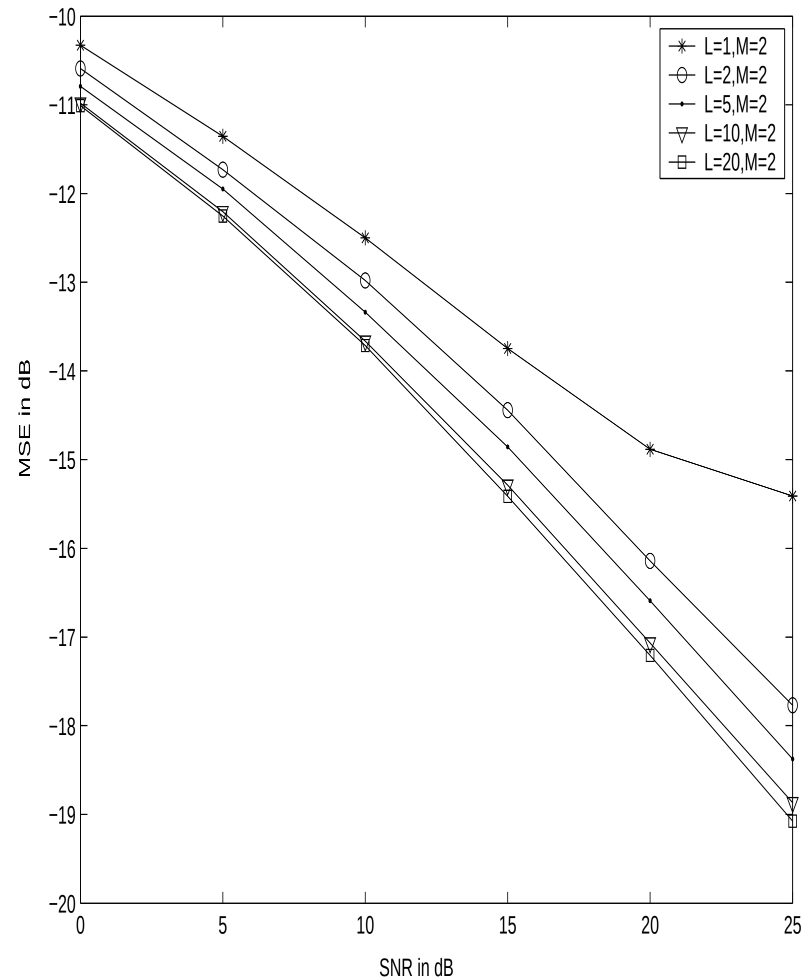
Effect of f_d @ SNR=15 dB

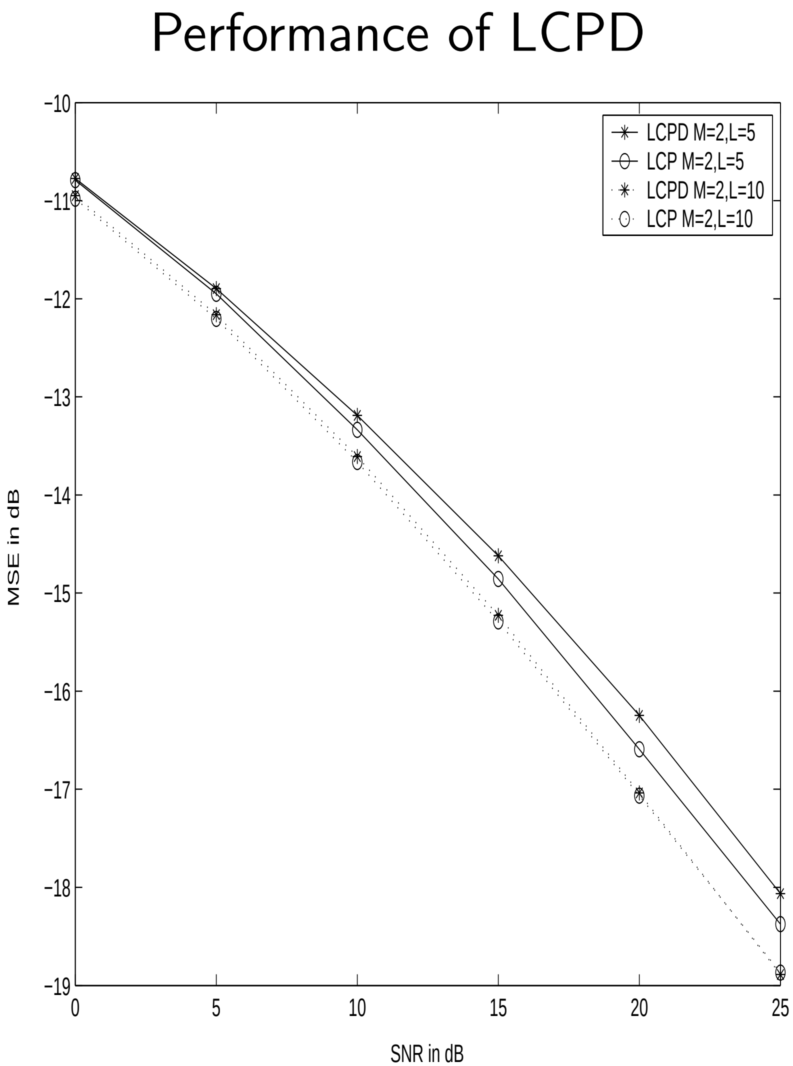


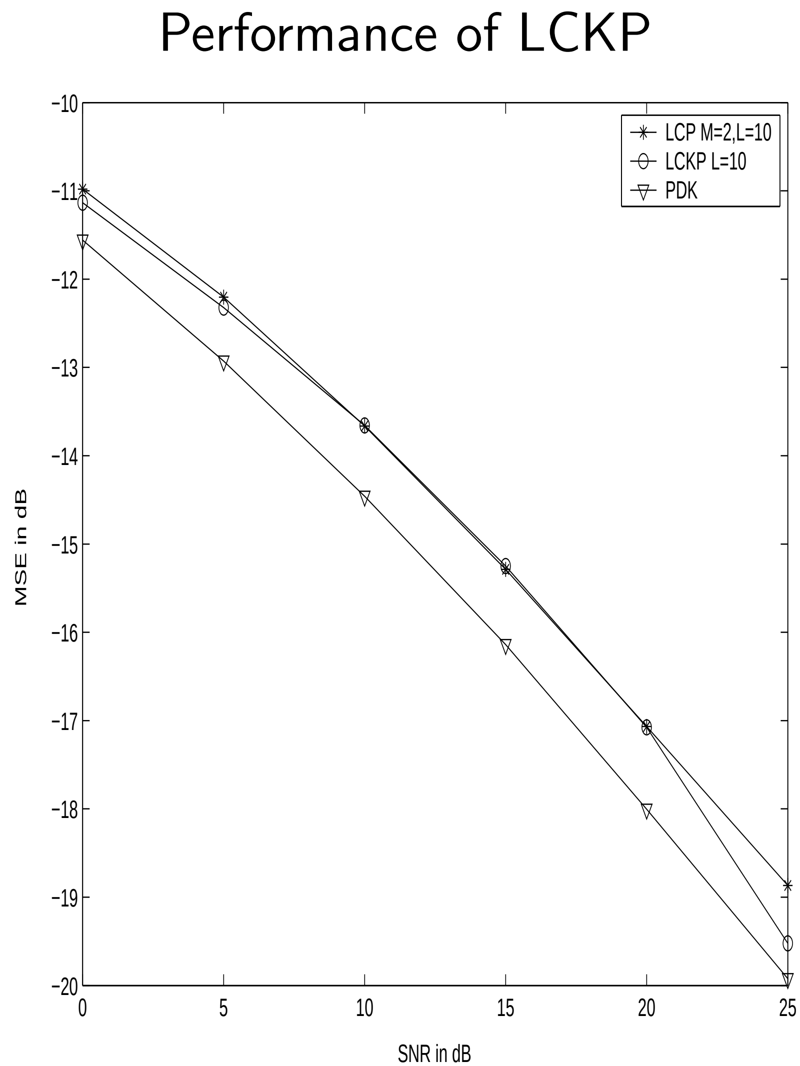
Effect of M @ $f_d = 0.01$



Effect of L @ $f_d = 0.01$




Performance of LCPD
LCPD M=2,L=5
LCP M=2,L=5
LCPD M=2,L=10
LCP M=2,L=10
MSE in dB
SNR in dB


Performance of LCKP
LCP M=2,L=10
LCKP L=10
PDK
MSE in dB
SNR in dB

Conclusions:

- Performance of pilot-only estimation is limited by training interval.
- Decision-directed Wiener and Kalman approaches require large matrix inversion.
- Proposed a low-complexity two-stage estimator:
  1. Computes smoothed past channel estimates.
  2. LTI Wiener prediction of the current channel (or steady-state Kalman prediction).

  Requires only small matrix inversion.
- Numerical results show that LCP performance is close to that of Kalman and Wiener estimators.