OFDM mmWave Channel Estimation with OMP

Wuqiong Zhao

This is a simple simulation of millimeter wave (mmWave) channel estimation in wideband assisted by OFDM with orthogonal matching pursuit (OMP) algorithm. The main idea of OFDM mmWave channel estimation is the shared angle of arrival (AoA) and angle of departure (AoD). However, in wideband mmWave MIMO systems, the beam squint effect cannot be neglected. For simplicity, this effect is not considered in this simulation. The number in the bracket after OMP is the number of carriers used to estimate the AoA and AoD (i.e. non-zero elements in the beam domain). The R suffix means re-estimating the carriers used to estimate the AoA and AoD using least square (LS) after the support is calculated.

Contents

1 System Settings

The simulation adopts the geometric channel model for millimeter wave (mmWave).

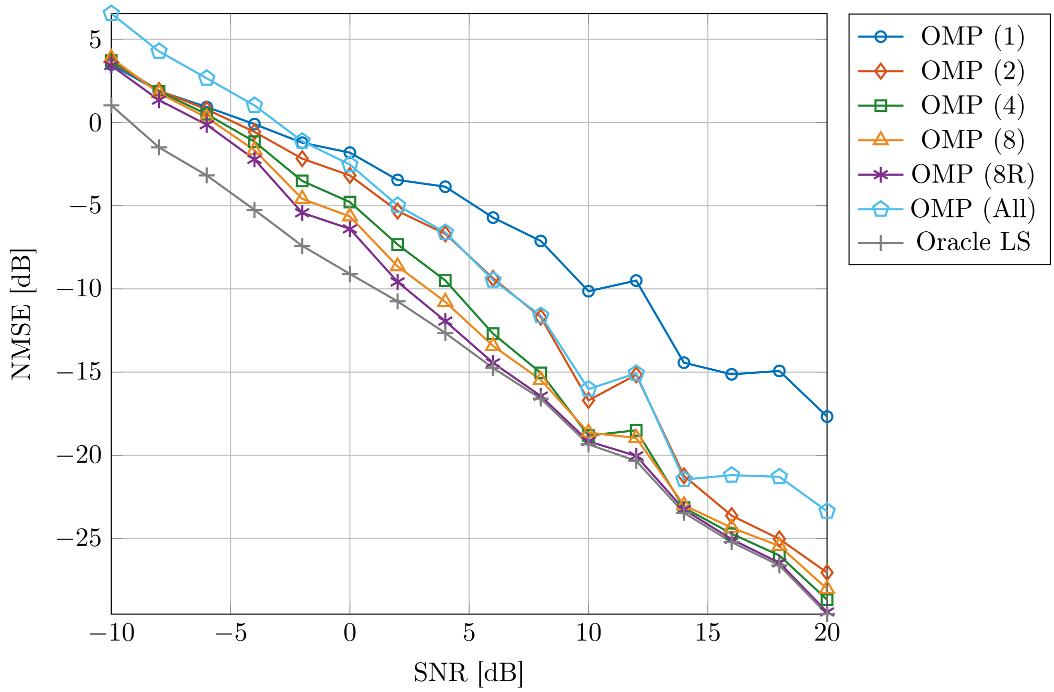
Name	Antenna Number	Beam Number	Grid Number
Transmitter	8×1	2×1	8×1
Receiver	16×1	4×1	16×1

- Channel Sparsity: 6;
- Off Grid Effect: false;
- Bandwidth: Wideband;
- Carriers: 64.

2 Simulation Results

2.1 NMSE v.s. SNR (Pilot: 16)

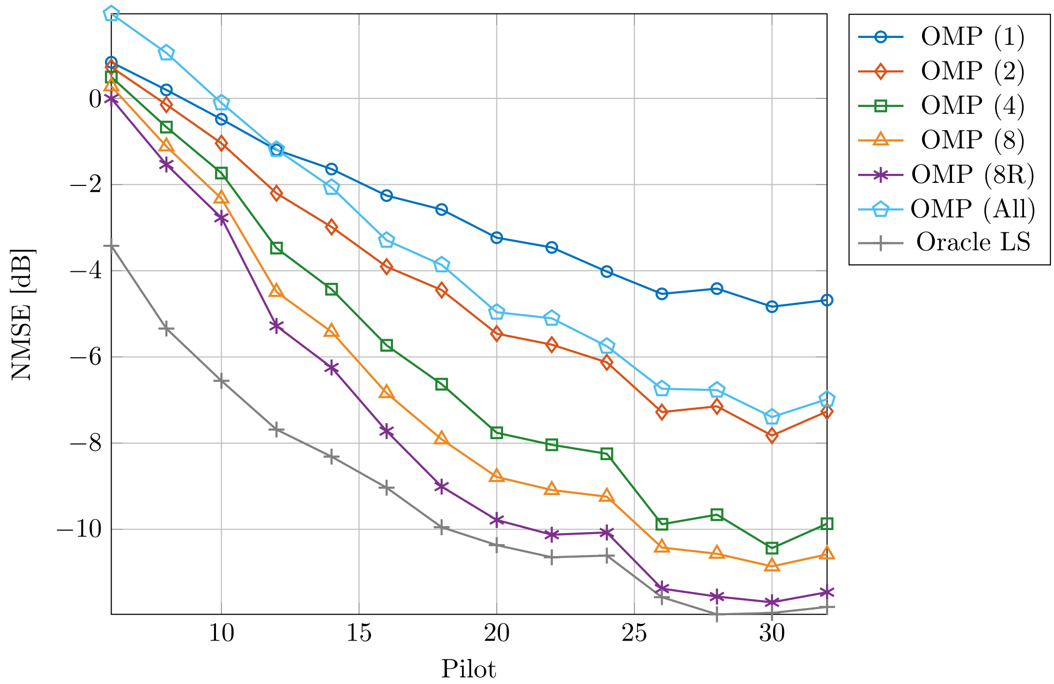
SNR [dB]	OMP (1)	OMP (2)	OMP (4)	OMP (8)	OMP (8R)	OMP (All)	Oracle LS
−10	3.47	3.63	3.72	3.87	3.42	6.55	1.03
−8	1.88	1.92	1.87	1.77	1.34	4.27	−1.50
−6	0.93	0.79	0.50	0.29	−0.14	2.66	−3.18
−4	−0.10	−0.57	−1.15	−1.66	−2.22	1.02	−5.25
−2	−1.22	−2.17	−3.51	−4.59	−5.43	−1.13	−7.41
0	−1.81	−3.18	−4.78	−5.65	−6.39	−2.53	−9.10
2	−3.45	−5.32	−7.33	−8.65	−9.58	−4.99	−10.75
4	−3.86	−6.69	−9.50	−10.78	−11.94	−6.61	−12.65
6	−5.72	−9.37	−12.69	−13.43	−14.44	−9.48	−14.75
8	−7.12	−11.70	−15.05	−15.46	−16.44	−11.60	−16.59
10	−10.14	−16.70	−18.82	−18.63	−19.16	−16.02	−19.34
12	−9.50	−15.16	−18.50	−18.96	−20.03	−15.08	−20.34
14	−14.43	−21.23	−23.15	−23.00	−23.26	−21.45	−23.45
16	−15.13	−23.63	−24.71	−24.35	−25.05	−21.19	−25.22
18	−14.93	−25.03	−26.02	−25.45	−26.46	−21.29	−26.62
20	−17.67	−27.05	−28.68	−28.03	−29.41	−23.36	−29.54



Simulated with 500 Monte Carlo tests.

2.2 NMSE v.s. Pilot (SNR: 0 dB)

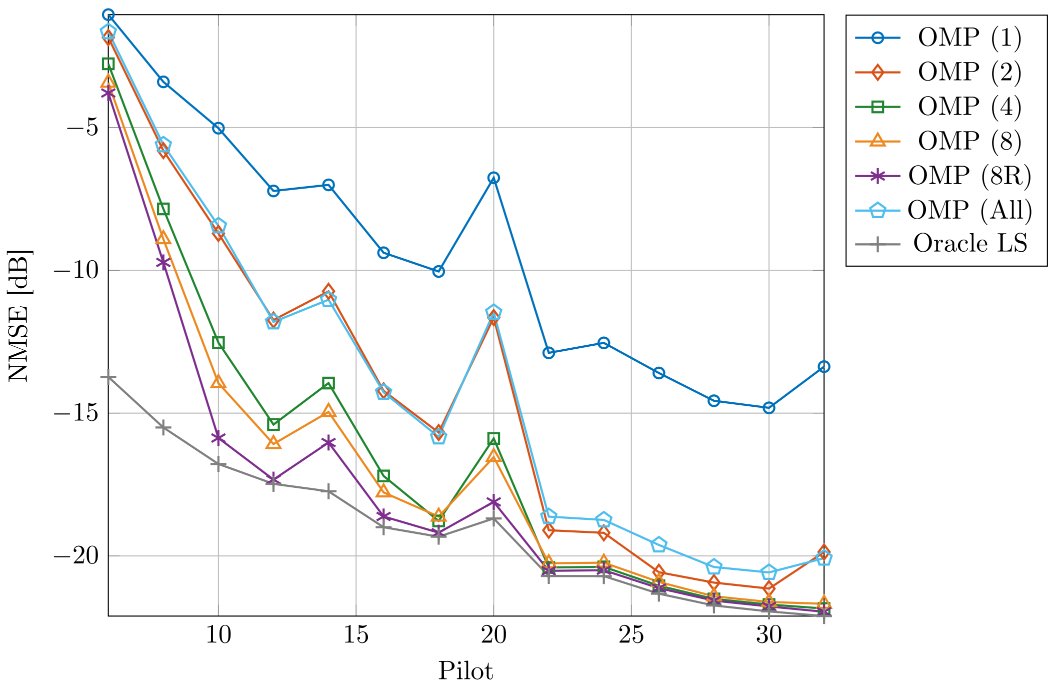
Pilot	OMP (1)	OMP (2)	OMP (4)	OMP (8)	OMP (8R)	OMP (All)	Oracle LS
6	0.84	0.72	0.50	0.28	0.00	1.96	−3.42
8	0.20	−0.14	−0.67	−1.11	−1.53	1.06	−5.34
10	−0.48	−1.04	−1.73	−2.32	−2.77	−0.10	−6.55
12	−1.19	−2.20	−3.47	−4.49	−5.28	−1.18	−7.68
14	−1.64	−2.98	−4.43	−5.42	−6.24	−2.06	−8.31
16	−2.26	−3.91	−5.73	−6.84	−7.72	−3.29	−9.03
18	−2.58	−4.45	−6.63	−7.91	−9.01	−3.86	−9.95
20	−3.23	−5.46	−7.76	−8.79	−9.78	−4.96	−10.37
22	−3.46	−5.71	−8.04	−9.09	−10.12	−5.10	−10.65
24	−4.02	−6.12	−8.25	−9.24	−10.07	−5.75	−10.61
26	−4.54	−7.28	−9.88	−10.42	−11.38	−6.74	−11.57
28	−4.41	−7.15	−9.66	−10.57	−11.56	−6.77	−11.97
30	−4.83	−7.82	−10.44	−10.86	−11.69	−7.39	−11.94
32	−4.68	−7.26	−9.87	−10.58	−11.46	−6.98	−11.80



Simulated with 500 Monte Carlo tests.

2.3 NMSE v.s. Pilot (SNR: 10 dB)

Pilot	OMP (1)	OMP (2)	OMP (4)	OMP (8)	OMP (8R)	OMP (All)	Oracle LS
6	−1.05	−1.85	−2.77	−3.43	−3.79	−1.65	−13.73
8	−3.40	−5.81	−7.85	−8.91	−9.72	−5.61	−15.50
10	−5.02	−8.71	−12.53	−13.95	−15.87	−8.44	−16.78
12	−7.22	−11.74	−15.40	−16.08	−17.33	−11.81	−17.48
14	−7.01	−10.73	−13.95	−14.96	−16.03	−11.03	−17.74
16	−9.38	−14.22	−17.20	−17.78	−18.62	−14.27	−19.00
18	−10.04	−15.68	−18.78	−18.62	−19.19	−15.83	−19.33
20	−6.76	−11.65	−15.89	−16.54	−18.11	−11.47	−18.69
22	−12.89	−19.10	−20.41	−20.26	−20.52	−18.62	−20.70
24	−12.54	−19.19	−20.38	−20.24	−20.50	−18.74	−20.71
26	−13.59	−20.57	−21.04	−20.91	−21.13	−19.62	−21.32
28	−14.57	−20.93	−21.51	−21.41	−21.56	−20.39	−21.73
30	−14.81	−21.14	−21.70	−21.61	−21.77	−20.58	−21.94
32	−13.37	−19.86	−21.84	−21.68	−21.95	−20.08	−22.11



Simulated with 500 Monte Carlo tests.

3 Simulation Configuration

3.1 Configuration File

Listing 1: `MIMO_wideband.sim`

```yaml
# MIMO_wideband.sim
# Wideband (OFDM) mmWave Channel Estimation with OMP
# Author: Wuqiong Zhao
# Date: 2022-09-26

version: 0.1.0 # the targeted mmCEsim version
meta: # document meta data
  title: OFDM mmWave Channel Estimation with OMP
  description:
    This is a simple simulation of millimeter wave (mmWave)
    channel estimation in wideband assisted by OFDM
    with orthogonal matching pursuit (OMP) algorithm.
    The main idea of OFDM mmWave channel estimation
    is the shared angle of arrival (AoA) and angle of departure
    (AoD). However, in wideband mmWave MIMO systems,
    the beam squint effect cannot be neglected. For simplicity,
    this effect is not considered in this simulation.
    The number in the bracket after OMP is the number of
    carriers used to estimate the AoA and AoD (i.e. non-zero
    elements in the beam domain). The R suffix means
    re-estimating the carriers used to estimate the AoA
    and AoD using least square (LS)
    after the support is calculated.
  author: Wuqiong Zhao
  email: contact@mmcesim.org
  website: https://mmcesim.org
  license: MIT
  date: "2022-09-16"
  comments: This is an uplink channel.
physics:
  frequency: wide # assume narrow band
  carriers: 64
  off_grid: false # do not consider off-grid problem
nodes:
  - id: BS # this should be unique
    role: receiver
    num: 1 # this is the default value
    size: [16, 1] # ULA with size 16*1
    beam: [4, 1]
    grid: same # the same as physics size
    beamforming:
      variable: "W"
      scheme: random
  - id: UE # user
    role: transmitter
    num: 1 # a single-user model
    size: 8 # ULA with size 8
    beam: 2
    grid: 8
    beamforming:
      variable: "F"
      scheme: random
channels:
  - id: H
    from: BS
    to: UE # 'from -> to' specifies the channel direction
    sparsity: 6
    gains:
```

```
      mode: normal
      mean: 0
      variance: 1
sounding:
  variables:
    received: "Y" # received signal vector
    noise: "noise" # received noise vector
    channel: "H_cascaded" # the cascaded channel (actually the same as 'H' for simple MIMO)
macro:
  - name: OFDM_ANGLE_EST_NUM
    value: 4
    in_alg: true
  - name: OFDM_RE_ESTIMATE
    value: false
    in_alg: true
  - name: SPARSITY_EST
    value: 6
    in_alg: true
preamble: |
  COMMENT Here starts the preamble.
estimation: |
  VNt::m = NEW `DICTIONARY.T`
  VNr::m = NEW `DICTIONARY.R`
  lambda_hat = INIT `GRID.*`
  H_hat = INIT `SIZE.R` `SIZE.T` `CARRIERS_NUM`
  Q = INIT `MEASUREMENT` `GRID.*`
  i::u0 = LOOP 0 `PILOT`/`BEAM.T`
    F_t::m = NEW F_{:,:,i}
    W_t::m = NEW W_{:,:,i}
    Q_{i*`BEAM.*`:(i+1)*`BEAM.*`-1,:} = \kron(F_t^T, W_t^H) @ \kron(VNt^*, VNr) # the sensing matrix
  END
  BRANCH
  angle_est = INIT `GRID.R`*`GRID.T` dtype=f
  k::u0 = LOOP 0 `OFDM_ANGLE_EST_NUM`
    none_zero::u1 = NEW \find(\abs(VNr^H@H_cascaded_{:,:,k}@VNt)>0.1)
    lambda_hat = ESTIMATE Q Y_{:,k} none_zero
    angle_est = angle_est + \pow(\abs(lambda_hat), 2)
    IF !`OFDM_RE_ESTIMATE`
      H_hat_{:,:,k} = VNr @ \reshape(lambda_hat, `GRID.R`, `GRID.T`) @ VNt^H
    END
  END
  ranking::u1 = NEW \sort_index(-angle_est)
  support::u1 = NEW ranking_{0:`SPARSITY_EST`-1}
  index_start::u0 = NEW 0
  IF !`OFDM_RE_ESTIMATE`
    index_start = `OFDM_ANGLE_EST_NUM`
  END
  k::u0 = LOOP index_start `CARRIERS_NUM`
    lambda_hat = CALL LS_support Q Y_{:,k} support
    H_hat_{:,:,k} = VNr @ \reshape(lambda_hat, `GRID.R`, `GRID.T`) @ VNt^H
  END
  RECOVER H_hat
  MERGE
conclusion: |
  PRINT "">>\t"" `JOB_CNT`+1 '/' `JOB_NUM` '\n'
simulation:
  backend: cpp # cpp (default) | matlab | octave | py
  metric: [NMSE] # used for compare
  jobs:
    - name: "NMSE v.s. SNR (Pilot: 16)"
      test_num: 500
      SNR: [-10:2:20]
```

```yaml
      SNR_mode: dB # dB (default) | linear
      pilot: 16
      # pilot_mode: percent # num (default) | percent
      algorithms: # compare different languages
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 1
          label: OMP (1) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 2
          label: OMP (2) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 4
          label: OMP (4) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 8
          label: OMP (8) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 8
            - name: OFDM_RE_ESTIMATE
              value: true
          label: OMP (8R) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 64
          label: OMP (All) # used in report
        - alg: Oracle_LS
          label: Oracle LS
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 64
    - name: "NMSE v.s. Pilot (SNR: 0 dB)"
      test_num: 500
      SNR: 0
      pilot: [6:2:32]
      # pilot_mode: percent # num (default) | percent
      algorithms: # compare different languages
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 1
          label: OMP (1) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 2
```

```yaml
          label: OMP (2) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 4
          label: OMP (4) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 8
          label: OMP (8) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 8
            - name: OFDM_RE_ESTIMATE
              value: true
          label: OMP (8R) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 64
          label: OMP (All) # used in report
        - alg: Oracle_LS
          label: Oracle LS
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 64
    - name: "NMSE v.s. Pilot (SNR: 10 dB)"
      test_num: 500
      SNR: 10
      pilot: [6:2:32]
      # pilot_mode: percent # num (default) | percent
      algorithms: # compare different languages
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 1
          label: OMP (1) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 2
          label: OMP (2) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 4
          label: OMP (4) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 8
          label: OMP (8) # used in report
        - alg: OMP
          max_iter: 6
```

```
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 8
            - name: OFDM_RE_ESTIMATE
              value: true
          label: OMP (8R) # used in report
        - alg: OMP
          max_iter: 6
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 64
          label: OMP (All) # used in report
        - alg: Oracle_LS
          label: Oracle LS
          macro:
            - name: OFDM_ANGLE_EST_NUM
              value: 64
  report:
    name: OFDM_mmWave_CE_OMP_Simulation
    format: [pdf, latex] # both compiled PDF and tex files
    plot: true # plot data
    table: false # do not print table
    latex:
      command: pdflatex # command to compile the report
      UTF8: false # no need for UTF8 support with this setting
```

3.2 Algorithms

Listing 2: OMP.alg

```
#! Function: OMP
#! Description: Orthogonal matching pursuit compressed sensing.
#! Author: Wuqiong Zhao
#! Date: 2022-09-16
#! Version: 0.1.0

# Input:
#   - Q: Sensing matrix
#   - y: Received signal
#   - L: Sparsity
# Output:
#   - h: The estimated sparse signal
h::v = FUNCTION OMP Q::m y::v L::u0
  COMMENT Start of OMP algorithm!
  h = \zeros(\size(Q, 1)) # initialize as zeros
  Q_H::m = NEW Q^H # the conjugate transpose of Q
  r = NEW y # residual
  r_last::v = NEW r * 2 # the residual in last iteration
  support = INIT \length(y) dtype=u # over-length support array
  term = INIT $\size(Q_H, 0)$ dtype=f # float number array
  j::u0 = NEW 0
  a::v = INIT
  FOR "" $j != \length(y)$ $j = j + 1$
    term = \abs(Q_H @ r)
    index::u0 = NEW \index_max(term)
    IF j && \ismember(index, support_{0:j-1})
      BREAK # end of the LOOP
    END
    support_{j} = index
    columns::m = NEW Q_{:, support_{0:j}}
    a = \solve(columns, y)
    r = y - columns @ a
    IF \sum(\abs(r - r_last)) / \sum(\abs(r_last)) < 0.0001 || j + 1 >= L
```

```
      j = j + 1
      BREAK # accurate enough to end iteration
    ELSE
      r_last = r
    END
  END
  # prepare for the final return
  h_{support_{0:j-1}} = a
END
```

Listing 3: `Oracle_LS.alg`

```
#! Function: Oracle_LS
#! Description: Oracle LS compressed sensing.
#! Author: Wuqiong Zhao
#! Date: 2022-09-18
#! Version: 0.1.0

# Input:
#   - Q: Sensing matrix
#   - y: Received signal
#   - indices: Indices of non-zero elements
# Output:
#   - h: The estimated sparse signal
h::v = FUNCTION Oracle_LS Q::m y::v indices::u1
  h = \zeros(\size(Q,1))
  h_{indices} = \pinv(Q_{:, indices}) @ y
END
```

4 mmCEsim Information

This report is auto generated by mmCEsim. The application **mmCEsim** is a powerful tool to simulate millimeter wave (mmWave) channel estimation (CE) for both experts and learners.

mmCEsim is *open source*! The software can be freely used and distributed under the MIT license.

- Official Website: `https://mmcesim.org`
- Documentation: `https://mmcesim.org/doc`
- Tutorial: `https://mmcesim.org/tutorial`
- Examples: `https://mmcesim.org/example`
- Web Application: `https://app.mmcesim.org`
- Blog: `https://blog.mmcesim.org`
- Publications: `https://pub.mmcesim.org`
- GitHub Organization: `https://github.com/mmcesim`
- Twitter: `https://twitter.com/mmcesim`
- VS Code Extension: `https://marketplace.visualstudio.com/items?itemName=mmcesim.mmcesim`