Technische Universität Dortmund

Stochastic Optimal Power Flow using LinDistFlow

Submitted to
Prof. Dr.-Ing. Timm Faulwasser
Institute of Control Systems
Hamburg University of Technology

for the fulfillment of the Project requirement

By:
Abhisek Adhikari (270463)
Sustainable Energy Systems
ETIT
Technische Universität Dortmund

October, 2025

ABSTRACT

New residential electrical loads such as Electric Vehicles, Heat Pumps have increased the uncertainty in power demand in the distribution network. This uncertainity challenges the formulation of deterministic optimal power flow problems. To address this, a stochastic optimal power flow model is developed using LinDistFlow approximation of the actual AC Power Flow model where random variables account for the load's uncertainties. The stochastic optimal power flow is solved using Polynomial Chaos Expansion, which utilizes orthogonal polynomial basis corresponding to the underlying probability distributions. The load uncertainity specified was in two load groups: Uniformly distributed load groups is modeled using Legendre Polynomial basis, and Gaussian distributed load group is modeled using Hermite Polynomial basis. This formulation transforms the stochastic problem into a deterministic optimization problem involving Polynomial Chaos Expansion coefficients. It is implemented in MATLAB/CasADi and is solved with Interior Point Optimizer (IPOPT) solver. The study is conducted on a custom 33-bus radial distribution network, and provides expected generator operating set points, evaluates the generator’s output sensitivity to demand variations, and confirms the voltage magnitude stays within the specified chance constraint. The result confirms that the deterministic PCE coefficients efficiently capture the stochastic behaviour of loads and provides a analytical viewpoint for optimal power flow analysis.

TABLE OF CONTENT

1 INTRODUCTION

Increasing integration of new residential loads and distributed energy sources such as electric vehicles, heat pumps, and small-scale renewable generation in the distribution grid has brought significant changes to the operational dynamics of the modern distribution grid. The new loads introduce high variability in power demand that challenges the application of deterministic optimal power flow formulations. The deterministic optimal power flow assumes fixed load conditions which does not account for the uncertainity in the loads present in real-world systems. Solving such deterministic problems without accounting for the uncertainity can result in sub optimal operating points. Accurate modeling for uncertainity in the OPF problems is required to ensure reliable, economic, voltage stable operation on the modern distribution grid.

Uncertainty modeling in power systems is done using random variables. The laws of power balance must be respected in all realizations of these random variables and this causes uncertainity in demand to propagate to other system quantities such as power generation and voltage magitudes. Consequently, these variables are also modeled using random variables. Several computation methods exist to incorporate randomness into optimal power flow problems. Monte Carlo simulation is a sampling based method which repeatedly draws samples of random quantity from its probability distribution and solves for the power balance to realize other system variables and estimates the statisitics of the unknown random variables. This approach is computationally expensive for distribution grids which have large number of nodes and branches. Moment based methods translate the random variables such as uncertain loads into statistical moments, such as mean and variance. These methods often rely on first order moment utilization and neglecting higher order moments for non linear power flow problems lead to loss of accuracy in modeling of the unknown random variables. These random variables live in a space described by their probability distributions. Polynomial Chaos Expanison uses orthogonal basis functions to span the space where these random variables live. The choice of orthogonal basis depends on the underlying probability distribution of the random variables. It reformulates the stochastic problem into a deterministic problem involving poynomial chaos expansion coefficients which are real numbers. Thus, use of Polynomial Chaos Expanison coefficients to model the random variables makes the stochastic opf problem computationally less expensive, and accurate with correct choice of basis functions.

This project work develops a stochastic optimal power flow formulation for a custom 33-bus radial distribution network using LinDistFlow approximation. The stochasticity in our problem description is specified on the demand side. Two different load groups are specified in problem description. Loads in first load group follow uniform distribution while the other load group follow gaussian distribution. The space for uniformly distributed load is spanned using Legendre polynomial basis and the space for gaussian load group is spanned with hermite polynomial basis with respective polynomial basis coefficients. These coefficients are real valued quantities and thus the resulting problem now is a deterministic optimization problem. It is implemented in MATLAB/CasADi and solved using Interior Point Optimizer (IPOPT). The resulting polynomial chaos expansion coefficients yield the expected generator set-points and quantify their sensitivity to load variations. The primary advantage of polynomial chaos expansion for stochastic OPF is its ability to accurately handle equality constraints that involve random varaibles and also help enforcing inequality constriants using moment based reformulations of chance constraints.

The remainder of the report is organized as follows. Section 2 summarizes the related work on stochastic and polynomial chaos based OPF formulations. Section 3 presents the mathematical formulation of our stochastic OPF problem and modeling of load uncertainity. Section 4 describes the implementation procedure adopted for CasADi implementation and discusses the structure of the deterministic problem. Section 5 presents and analyzes the numerical results and plots. Finally, section 6 concludes the report and outlines directions of further work.

2 Related Work

Uncertainity modeling methods are generic and not specifically designed to handle uncertainites in modern power systems. Uncertainity modeling in power systems is just one of the application scenarios of such uncertainity modeling methods. According to Wang *et al.* [1], Monte Carlo simulation is grouped under simulation based methods, which estimate statistical quantities through repeated random realizations of uncertain inputs. It groups

Polynomial Chaos Expansion (PCE) into broader group of surrogate model methods, which constructs analytical representation of stochastic dependencies using orthogonal polynomial bases. According to Roald *et al.* [2], uncertainity in power system can be divided into input uncertainity and output uncertainity. Input uncertainty originates from stochastic variations in generations, loads, or even network parameters and enters directly into model equations. Ouput uncertainity arises from resulting quantities such as voltages, generation costs, etc. Even with a perfect estimate of uncertainity from each source, it can be challenging to determine how uncertainties from multiple sources combine and propagate. This difficulty propagates even for linear systems and becomes significantly complex for nonlinear AC power flow equations. Mühlpfordt *et. al* [3] show that PCE for stochastic OPF can be used for full AC power flow model, they propose a chance constrained AC OPF approach using Polynomial Chaos Expansion to propagate uncertainties through nonlinear power flow equations.

3 Problem Statement

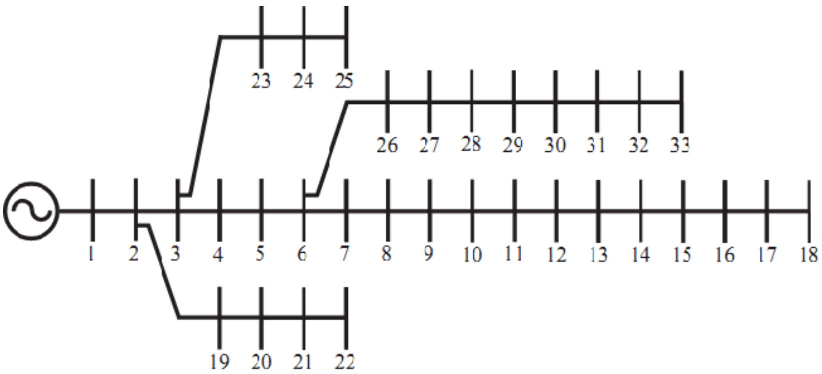

Figure 1

The network considered in this project work is a 33-bus radial distribution system operating at 12.66 kV. The single line diagram of the distribution grid is shown in figure 1. Node 1 is the slack node with a fixed voltage magnitude 1.0 p.u., and it is connected to the transmission grid. Node 16 and 32 represent the additional generator buses. All other nodes serve as the load buses consuming both active and reactive power.

The active power demand at each load is uncertain and hence modeled as a random variable. The reactive power demand is treated as a deterministic quantity and the value is directly taken from the MATPOWER data. For simplicity the reactive power of non slack generators are fixed at zero, so it concludes all the reactive power is to be provided by the slack bus. The reactive power can't be neglected as it is included in the voltage magnitude equations of the LinDistFlow model.

Two load groups are defined according to their probability distributions:

- Group 1 (Uniform loads): Buses 2-18 follow a uniform distribution.

$$P_{D,i} \sim U(0.95 P_{D,i}^{mat}, 1.05 P_{D,i}^{mat})$$

- Group 2 (Gaussian loads): Buses 19-33 follow a Gaussian distribution.

$$P_{D,i} \sim N(P_{D,i}^{mat}, \left(0.05 P_{D,i}^{mat}\right)^2)$$

The objective of the stochastic optimal power flow problem is to minimize the expected generation cost given by

$$f = \mathbb{E}\left[\sum_{i \in G} (c_{2,i} P_{G,i}^2 + c_{1,i} P_{G,i} + c_{0,i})\right]$$

subject to:

- Power balance and voltage drop equations from LinDistFlow
- Due to uncertain nature of load it is highly unlikely that any inequalities involving grid quantities characterized by a random variable Z can be satisfied at all times with all the realizations of load so for inequality constraints the random variable Z is formulated as a chance constraint and is satisfied with 95% probability or 5% constraint violation probability. ε is the constraint violation probability.

$$\Pr[z_{min} \leq Z \leq z_{max}] \geq 1 - \varepsilon, \varepsilon = 0.05$$

LinDistFlow is model that is linear in active and reactive powers and relates bus voltages to power flows through line parameters. Incorporating uncertain loads into the LinDistFlow equations leads to propagation of uncertainty to power injections at nodes, power flows at branches and voltage magnitudes at the nodes. Thus, the uncertain active power demands $P_{D,i}$ are expanded using Polynomial Chaos Expansion (PCE) orthogonal basis depending on their probabilistic distribution. This allows all other uncertain quantities to be also represented using the weighted combination of PCE orthogonal basis.

4 Models and Mathematical Problem Formulation

4.1 LinDistFlow Model

Power flow in 33 bus radial distribution grid is modeled using LinDistFlow. It is derived from the DistFlow model by dropping the quadratic loss terms in the power balance and voltage equations. LinDistflow provides a linearized active and reactive power balance and a linear relationship between nodal voltages and nodal power injections.

For a line connecting node i to node j, the linearized active and reactive power balances are expressed as:

$$P_{ij} = P_{D,j} - P_{G,j} + \sum P_{jk}$$

$$Q_{ij} = Q_{D,j} - Q_{G,j} + \sum Q_{jk}$$

The voltage drop along the line is approximated by:

$$V_i - V_j = R_{ij}P_{ij} + X\_ijQ_{ij}$$

where

- $P_{ij,}$ Q_{ij} = active and reactive power flows on the branch (i,j)
- $R_{ij,}$ X_{ij} = branch resistance and reactance for the branch (i,j)
- $P_{D,j,}$ $Q_{D,j}$ = active and reactive demands at node j
- $V_{i,}$ V_j = voltage magnitudes at the sending and receiving nodes

The linear system relates the bus voltages to nodal power injections as

$$\Delta V = RP_{inj} - XQ_{inj}$$

where

- $P_{inj} = P_G - P_D$
- $Q_{inj} = Q_G - Q_D$

4.2 Polynomial Chaos Expansion

A independent random variable driving the uncertainity if it has finite variance then it lives within the Hilbert Space and can be expanded with deterministic PCE coefficients.

A random variable Z can be expanded in truncated PCE as:

$$\mathrm{Z} = \sum_{k=0}^{L-1} \mathrm{a_k} \Phi_\mathrm{k}$$

where a_k for k = 0 to L-1 are the determinsitic PCE coefficients. Φ_k are the orthogonal basis functions.

The basis functions must strictly follow orthogonality:

$$\mathbb{E}\left[\Phi_i \Phi_j\right] = 0 \; for \; i \neq j$$

The mean and variance follow:

$$\mathbb{E}[Z] = a_0$$

$$\mathrm{Var}[Z] = \sum_{k=1}^{L-1} \mathrm{a_k^2} < \Phi_\mathrm{k} , \Phi_k >$$

$<\Phi_k, \Phi_k>$ is the norm of the basis function.

In our problem we have two uncertain load groups. Loads in node 2 to 18 are distributed uniformly

$$P_{D,i} \sim U(0.95 P_{D,i}^{mat} , 1.05 P_{D,i}^{mat})$$

and loads in nodes 19 to 33 follow gaussian distribution:

$$P_{D,i} \sim U(0.95 P_{D,i}^{mat} , 1.05 P_{D,i}^{mat})$$

We model these two uncertain load groups using two different orthogonal polynomial bases:

- ζ for the uniform load group, and
- ξ for the gaussian load group

We use distribution matched orthogonal polynomial bases:

- Legendre for uniform, with $\zeta \sim U(-1,1)$ and $\Phi(\zeta) = \zeta$
- Hermite for gaussian, with $\xi \sim N(0,1)$ and $\Phi(\xi) = \xi$

U(0,1) could also be used but under U(-1,1) the Legendre polynomial basis are orthogonal and using U(0,1) would require using of shifted basis for Legendre polynomial to maintain orthogonality hence U(-1,1) is chosen.

When

$$P_{D,i} \sim U(0.95P_{D,i}^{mat}\ ,1.05P_{D,i}^{mat})$$

$$h = \frac{1.05P_i - 0.05P_i}{2}$$

h results to $0.05P_i$

Thus the uniformly distributed load can be represented as $P_{D,i} = \mu + h\zeta$

Choice of ζ given by U(-1,1) shows when ζ is -1:

$P_{D,i} = P_i + 0.05P_i * \zeta$, we get $P_{Di} = 0.95P_i$ and

when ζ is 1 we get $P_{Di} = 1.05P_i$

Choice of ζ also needs to implement the norm of basis to be $< \Phi_\zeta, \Phi_\zeta >= \frac{1}{3}$

So we model the uncertainity in our problem using three polynomial bases:

- The zeroth order polynomial basis given by Φ_0
- The polynomial basis for the unformly distributed loads Φ_ζ
- The polynomial basis for the gaussian loads Φ_ξ

With two independent random variables one for uniformly distributed loads and other for gaussian loads and with first order polynomial chaos expansion any random variable can be expressed as:

$$Z(\zeta, \xi) = Z_0 + Z_\zeta \Phi(\zeta) + Z_\xi \Phi(\xi)$$

Our uncertain demand can also be expressed as:

$$P_D(\zeta, \xi) = P_{D,0} + P_{D,\zeta} \Phi(\zeta) + P_{D,\xi} \Phi(\xi)$$

4.3 Propagation of uncertainity from P_D to P_G and V via LinDistFlow

Power balance must be respected at all cases in power system. When the loads are described as random variable $P_D(\zeta, \xi)$

- $P_{inj} = P_G - P_D(\zeta,\xi)$

- $Q_{inj} = Q_G - Q_D$, the reactive loads are specified to be deterministic with Q_D deterministic and $Q_G = 0$ for non slack generator buses.

LinDistFlow gives linear relationship between power injections and voltages:

$$V(\zeta,\xi) = V_{ref} - RP_{inj}(\zeta,\xi) + XQ_{inj}$$

P_G, P_D and V are quantiteis of interest for us, these three quntities can be expanded in the same basis as:

$$P_D(\zeta,\xi) = P_{D,0} + P_{D,\zeta}\Phi(\zeta) + P_{D,\xi}\Phi(\xi)$$
$$P_G(\zeta,\xi) = P_{G,0} + P_{G,\zeta}\Phi(\zeta) + P_{G,\xi}\Phi(\xi)$$
$$V(\zeta,\xi) = V_0 + V_\zeta\Phi(\zeta) + V_\xi\Phi(\xi)$$

where

$$V_0 = V_{ref} - R(P_{G,0} - P_{D,0)} - X(Q_G - Q_D)$$
$$V_\zeta = -R(P_{G,\zeta} - P_{D,\zeta})$$
$$V_\xi = -R(P_{G,\xi} - P_{D,\xi})$$

The uncertainity in loads propagate to P_G and V through the PCE coefficients.

4.4 Expected Cost

The stochastic objective is given as:

$$f = \mathbb{E}\left[\sum_{i\in G}(c_{2,i}P_{G,i}^2 + c_{1,i}P_{G,i} + c_{0,i})\right]$$

which uses

$$\mathbb{E}\left[P_{G,i}^2\right] = P_{G,i,0}^2 + \frac{1}{3}P_{G,i,\zeta}^2 + \ P_{G,i,\xi}^2$$

because

$$\mathbb{E}[Z^2] = \mathbb{E}[\mathrm{Z}]^2 + Var[Z]$$

and

$$\mathrm{Var}[Z] = \sum_{k=1}^{L-1} \mathrm{a}_\mathrm{k}^2 < \Phi_\mathrm{k}\,, \Phi_k >$$

for Legendre polynomial basis of U(-1,1), $< \Phi_k, \Phi_k >= \frac{1}{3}$

for Hermite polynomial basis of N(0,1), $< \Phi_k, \Phi_k >=1$

The resulting cost function is:

$$f = S_b^2 c_2^T \left(P_{G,i,0}^2 + \frac{1}{3} P_{G,i,\zeta}^2 + P_{G,i,\xi}^2 \right) + S_b c_1^T P_{G,0}$$

4.5 Chance constraints and SOC formulation

For any random variable composed of legendre and hermite polynomial basis as $Z(\zeta, \xi) = Z_0 + Z_\zeta \Phi(\zeta) + Z_\xi \Phi(\xi)$,

$$Var[Z] = \frac{1}{3} Z_\zeta^2 + Z_\xi^2$$

When we implement two sided 95% bounds with a violation probability of ε = 0.05 we get λ(ε)= 1.96 from the standard score

- $V_{min} \leq V_0 - \lambda \sqrt{Var[V]}$
- $V_0 + \lambda \sqrt{Var[V]} \leq V^{max}$
- $P_{Gmin} \leq P_{G,0} - \lambda \sqrt{Var[P_G]}$
- $P_{G,0} + \lambda \sqrt{Var[P\_G\ ]} \leq P_G{}^{max}$

These inequalities can be be reformulated as Second Order cone as:

$$||Ax + b|| \leq c^T x + d$$

The resulting optimization problem is of following characteristic:

- Dimension of PCE : 3
- Objective function: Convex quadratic in the PCE coefficients
- Equalities: Linear because of the LinDistFlow model
- Inequalities: Second Order Cones because of chance constraint formulation

5 Results and Discussion

5.1 Polynomial Chaos Coefficients

The proposed PCE based stochastic optimal power flow problem was implemented in MATLAB R2024b/CasADi 3.6.7 and solved with Interior Point Optimizer 3.14.11 (IPOPT). The base power is S_b = 100MVA for a custom 33-bus radial distribution network. The optimization solution provides deterministic PCE coefficients for each generator's active power output:

$$P_{G,i}(\zeta,\xi) = P_{G,0,i} + P_{G,\zeta,i}\Phi(\zeta) + P_{G,\xi,i}\Phi(\xi)$$

For the non slack generators the obtained PCE coefficients are:

Bus	PG_0 (p.u.)	$P_{G\zeta}$ (p.u.)	$P_{G\xi}$ (p.u.)
16	0.237682	0.002052	0.003014
32	-0.024636	0.004105	0.006072

The zeroth order P_{G0} represents the expected/mean generation level, whereas the first order coefficients $P_{G\zeta}$ and $P_{G\xi}$ quantify the sensitivity of generator output to the variations in uniform and Gaussian loads.

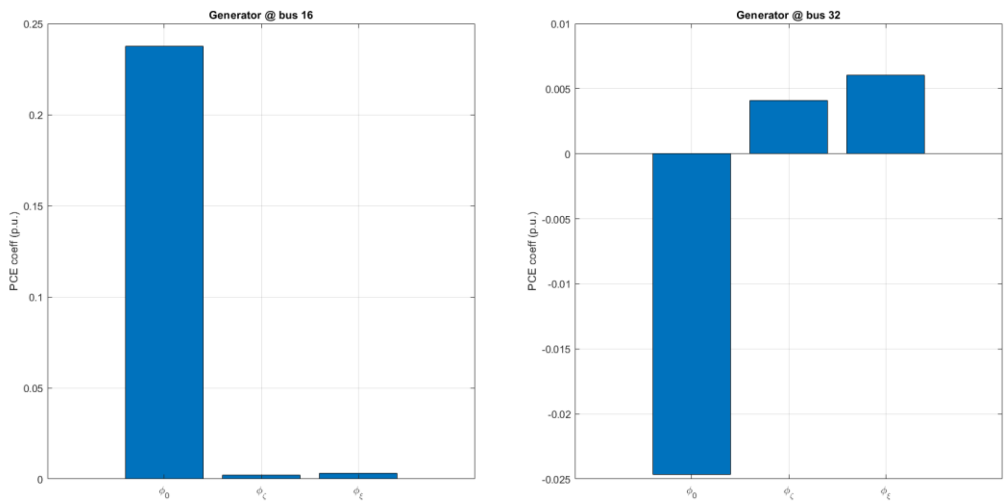

Figure 2 Generator 16 and 32 PCE Coefficients

5.2 Voltage Sampling

Using the computed PCE coefficients for the voltage magnitude, 100 samples of voltage at node 18 were generated using random realizations of ζ and ξ.

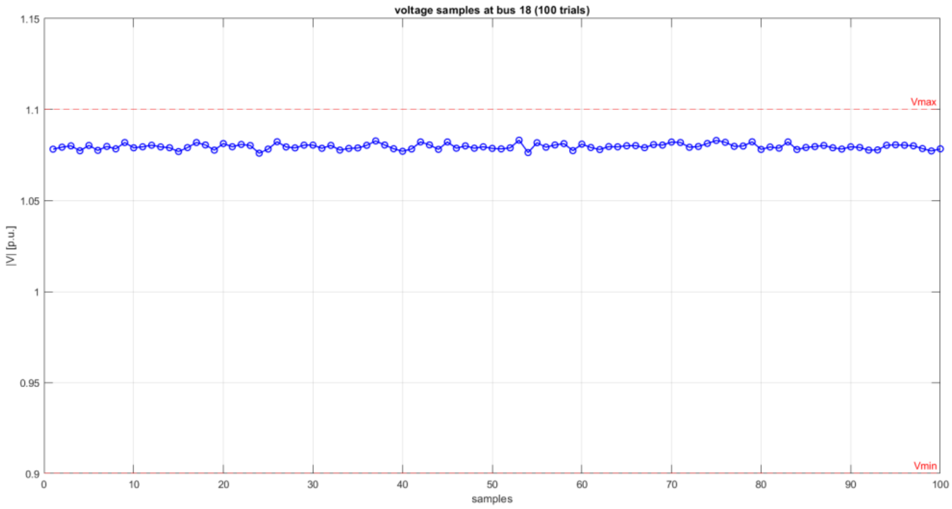

Figure 3 voltage samples at node 18 (100 samples)

The voltage at node 18 can be expressed in PCE coefficient as:

$$V_{18}(\zeta, \xi) = V_{18,0} + V_{18,\zeta}\Phi(\zeta) + V_{18,\xi}\Phi(\xi)$$

Clearly, not a single voltage sample violate the voltage limit at V_{min} = 0.90 and V_{max} =1.10. This might be due to relaxed 5% violation probability and also choice of value for λ from the standard z score. Same value of λ was used for uniformly distributed loads where there is influence of both uniform and Gaussian distributions and the demand standard deviation was taken to be only 0.05 or 5% as decribed in Gaussian loads and also Uniformly distributed loads varied just bewtween 0.95 and 1.05 of nominal power which might not be sufficient to violate the voltage limits.

5.3 Convergence and Computational Remarks

The optimization problem solved using Interior Point Optimizer Mehtod 3.14.11 adn NUMPS 5.4.1 linear solver successfully converged to an optimal solution after 75 iterations. The computational time was 0.222 seconds for a Intel i5 CP @ 2.50 GHz.

6 References

[1] Z. Wang, X. Lu, S. Zhang, and Y. Xu, “A comprehensive review on uncertainty modeling methods,” *Applied Energy*, vol. 360, 2025, Art. no. 122412

[2] L. Roald, J. Warrington, and G. Andersson, “Power Systems Optimization under Uncertainty,” *Proceedings of the IEEE*, vol. 111, no. 2, pp. 163–186, Feb. 2023

[3] T. Mühlpfordt, L. Roald, V. Hagenmeyer, T. Faulwasser, and S. Misra, “Chance-Constrained AC Optimal Power Flow — A Polynomial Chaos Approach

7 Appendix: MATLAB Code Problem Formulation and Plots

```
function result = stochastic_opf(mpcase, demand_std, lambda)

import casadi.*;

define_constants;

%% get number of buses, lines, and generators as in exercise 4
[n_buses, ~] = size(mpcase.bus);
[n_lines, ~] = size(mpcase.branch);
[n_gens,  ~] = size(mpcase.gen);
Sb = mpcase.baseMVA;

%% getting generator buses and also locating slack bus
g_buses   = mpcase.gen(:, GEN_BUS);
bus_type  = mpcase.bus(:, BUS_TYPE);
slack_bus = mpcase.bus(bus_type==3, BUS_I);
if isempty(slack_bus)
    error('No slack bus found.');
end

%% Active Power Demands in p.u.
%% Reactive power Demands in p.u.

Pd_mat = mpcase.bus(:, PD)/Sb;
Pd_mat(~isfinite(Pd_mat)) = 0;
Qd_mat = mpcase.bus(:, QD)/Sb;
Qd_mat(~isfinite(Qd_mat)) = 0;

%% we just take non zero demands
d_mask  = Pd_mat ~= 0;
d_buses = mpcase.bus(d_mask, BUS_I);
Pd_nz   = Pd_mat(d_mask);
Qd_nz   = Qd_mat(d_mask);
```

```
n_demands = numel(d_buses);

% mapping generator to buses
Cg = sparse(g_buses, 1:n_gens, 1, n_buses, n_gens);

% mapping demands to buses
Cd = sparse(d_buses, 1:n_demands, 1, n_buses, n_demands);

%% for dist flow getting the network topology data
F = mpcase.branch(:, F_BUS);
T = mpcase.branch(:, T_BUS);
R = mpcase.branch(:, BR_R);
R(~isfinite(R)) = 0;
X = mpcase.branch(:, BR_X);
X(~isfinite(X)) = 0;

%% Incidence matrix formation A branchbybus 32*33
A = sparse((1:n_lines)', F,  1, n_lines, n_buses) + sparse((1:n_lines)', T, -1, n_lines, n_buses);

%% deterministic Q branch flows
q_inj = -sparse(d_buses, 1, Qd_nz, n_buses, 1); % taking q injection as negtive as they are loads
AA = A*A';

% its just matrix inversion to get Qbranchflow but A is non square 33*32
% matrix so inverse wont exist in matlab domain hence used ATAx=ATb logic.

Qbr_det = full( (AA + 1e-9*speye(size(AA))) \ (A*q_inj) );

Qbr_det(~isfinite(Qbr_det)) = 0;

%% PCE series dimension = 3
L_pce = 3;  % [phi0, phi_zeta, phi_xi]
Duniform = intersect((2:18).', d_buses);
Dgauss   = setdiff(d_buses, Duniform);

Pd_coeff = zeros(n_demands, L_pce); % for all non zero demands we use 3 basis so 3 columns

for k = 1:n_demands
   b = d_buses(k);
```

```
    mu = Pd_nz(k);
    if ismember(b, Duniform)
        Pd_coeff(k,:) = [mu, demand_std*mu, 0]; % coeffecient for  zeta uniform dist

    else
        Pd_coeff(k,:) = [mu, 0, demand_std*mu]; %  coefficient for xi gaussian dist

    end
end

%% Getting voltage limits
Vmin = mpcase.bus(:, VMIN);
Vmax = mpcase.bus(:, VMAX);

Vmin2 = Vmin.^2;
Vmax2 = Vmax.^2;

%% Getting active power generation limits
Pg_min = mpcase.gen(:, PMIN)/Sb;

Pg_max = mpcase.gen(:, PMAX)/Sb;



%% Creating a stochastic LinDistFlow OPF problem
opti = casadi.Opti();

%% Defining varaibles as coefficients of PCE series
PG  = opti.variable(n_gens,  L_pce);   % generator PCE coeffs in 3 basis 1, zeta and xi
Pbr = opti.variable(n_lines, L_pce);   % branch P flows PCE coeffs same
V   = opti.variable(n_buses, L_pce);   % voltage-squared PCE coeffs same

%% Slack voltage-squared, slack bus is the reference fixing it at 1.0 p.u. regardless

opti.subject_to(V(slack_bus,1) == 1);
opti.subject_to(V(slack_bus,2) == 0);
opti.subject_to(V(slack_bus,3) == 0);

%% Power balance for all basis AT*Pbranch= Cg*Pg-Cd*Pd
for col = 1:L_pce
```

```
    opti.subject_to( A' * Pbr(:,col) == Cg*PG(:,col) - Cd*Pd_coeff(:,col) );
end

%% Voltage drop equality constraint for  each line from 1 to n_lines
%% Vend = Vsource - (R*P+X*Q)
for l = 1:n_lines
    i = F(l); j = T(l);
    opti.subject_to( V(j,1) == V(i,1) - 2*( R(l)*Pbr(l,1) + X(l)*Qbr_det(l) ) );
    opti.subject_to( V(j,2) == V(i,2) - 2*( R(l)*Pbr(l,2) ) );
    opti.subject_to( V(j,3) == V(i,3) - 2*( R(l)*Pbr(l,3) ) );
end

%%  Chance constraint formulation for Pgen and Voltage
k2 = lambda^2;

%% Basis norms for PCE Legendre ζ on [-1,1], Hermite ξ on N(0,1)
phi_norms = [1, 1/3, 1];  % <phi0^2>, <phi_zeta, phi_zeta>, <phi_xi, phi_xi>

%% Formulating the second-order cone

varPG = phi_norms(2)*PG(:,2).^2 + phi_norms(3)*PG(:,3).^2;
varV  = phi_norms(2)*V(:,2).^2  + phi_norms(3)*V(:,3).^2;

% for active power of generator

opti.subject_to( k2*varPG <= (PG(:,1) - Pg_min).^2 );
opti.subject_to( k2*varPG <= (Pg_max - PG(:,1)).^2 );




% for voltages of the lines

opti.subject_to( k2*varV  <= (V(:,1) - Vmin2).^2 );
opti.subject_to( k2*varV  <= (Vmax2 - V(:,1)).^2 );

%% Quadratic cost set up
c2 = mpcase.gencost(:, 5);
c1 = mpcase.gencost(:, 6);
E_PG2 = PG(:,1).^2 + phi_norms(2)*PG(:,2).^2 + phi_norms(3)*PG(:,3).^2;
```

```
cost = (Sb^2)*(c2.'*E_PG2) + Sb*(c1.'*PG(:,1));
opti.minimize(cost);

%% Initialize the solver

opti.set_initial(PG, 0);

opti.set_initial(Pbr, 0);
opti.set_initial(V(:,1), 1);
opti.set_initial(V(:,2:3), 0);

opti.solver('ipopt');

sol = opti.solve();

% get computation time
result.ex_time = max(sol.stats.t_wall_total, sol.stats.t_proc_total);

PG_star  = sol.value(PG);
Pbr_star = sol.value(Pbr);
V_star   = sol.value(V);

result.PG_pce  = PG_star;
result.V18_pce = V_star(18,:);

result.Pg_mean = PG_star(:,1)*Sb;
result.Pg_std  = sqrt( phi_norms(2)*(PG_star(:,2)*Sb).^2 + phi_norms(3)*(PG_star(:,3)*Sb).^2 );

result.pf_mean = Pbr_star(:,1)*Sb;
result.pf_std  = sqrt( phi_norms(2)*(Pbr_star(:,2)*Sb).^2 + phi_norms(3)*(Pbr_star(:,3)*Sb).^2 );

result.V2_mean = V_star(:,1);
result.V2_std  = sqrt( phi_norms(2)*V_star(:,2).^2 + phi_norms(3)*V_star(:,3).^2 );

v0 = V_star(18,1);
```

```
vZ = V_star(18,2);
vX = V_star(18,3);

N = 100;
zeta = -1 + 2*rand(N,1);
xi   = randn(N,1);

V2_samples = v0 + vZ.*zeta + vX.*xi;
V_samples  = sqrt(max(V2_samples,0));

Vmin = mpcase.bus(18,VMIN);
Vmax = mpcase.bus(18,VMAX);

viol = (V_samples < Vmin) | (V_samples > Vmax);
num_viol = sum(viol);
fprintf('Node 18 voltage violations: %d out of %d samples.\n', num_viol, N);
disp('Sampled voltages at node 18 (p.u.):');
disp(V_samples.');

% (optional) plot them
figure;
plot(1:N, V_samples, 'bo-', 'LineWidth', 1.2); hold on;
yline(Vmin, 'r--', 'Vmin');
yline(Vmax, 'r--', 'Vmax');
xlabel('samples'); ylabel('|V| [p.u.]');
title(sprintf('voltage samples at bus 18 (%d trials)', N));
grid on;




idx_g16 = find(g_buses == 16, 1);
idx_g32 = find(g_buses == 32, 1);
figure;
tiledlayout(1,2);
nexttile;
bar(PG_star(idx_g16,:));
xticklabels({'\phi_0','\phi_\zeta','\phi_\xi'});
ylabel('PCE coeff (p.u.)'); title('Generator @ bus 16');
```

```
grid on;

nexttile;

bar(PG_star(idx_g32,:));

xticklabels({'\phi_0','\phi_\zeta','\phi_\xi'});

ylabel('PCE coeff (p.u.)'); title('Generator @ bus 32');

grid on;



coeff_gen16 = PG_star(idx_g16,:);

coeff_gen32 = PG_star(idx_g32,:);

fprintf('Coefficients for generators at bus 16: PG0 = %.6f, PG_zeta = %.6f, PG_xi = %.6f \n',coeff_gen16(1),coeff_gen16(2),coeff_gen16(3));

fprintf('Coefficients for generators at bus 32: PG0 = %.6f, PG_zeta = %.6f, PG_xi = %.6f',coeff_gen32(1),coeff_gen32(2),coeff_gen32(3));



end
```