MULTI-PERIOD OPTIMAL POWER FLOW

By

ARYAN RITWAJEET JHA

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
School of Electrical Engineering and Computer Science

AUGUST 2026

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of ARYAN RITWAJEET JHA find it satisfactory and recommend that it be accepted.

Anamika Dubey, Ph.D., Chair

Committee Member A, Ph.D.

Committee Member C, Ph.D.

ACKNOWLEDGMENT

TBA

MULTI-PERIOD OPTIMAL POWER FLOW

Abstract

by Aryan Ritwajeet Jha, Ph.D.
Washington State University
August 2026

: Anamika Dubey

TBA

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Dedication

TBA

Chapter One

SOME FORMATTING EXAMPLES

1.1 Chapter one tittle section

Chapter Two

FIGURES AND TABLES

2.1 Examples of a figure

REFERENCES

[1] Aayushya Agarwal and Larry Pileggi. “Large Scale Multi-Period Optimal Power Flow With Energy Storage Systems Using Differential Dynamic Programming”. In: *IEEE Trans. Power Syst.* 37.3 (Sept. 2021), pp. 1750–1759. ISSN: 1558-0679. DOI: 10.1109/TPWRS.2021.3115636.

[2] Masoud Farivar and Steven H. Low. “Branch flow model: Relaxations and convexification”. In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)* (Dec. 2012), pp. 3672–3679. ISSN: 0743-1546. DOI: 10.1109/CDC.2012.6425870.

[3] Nawaf Nazir and Mads Almassalkhi. “Receding-Horizon Optimization of Unbalanced Distribution Systems with Time-Scale Separation for Discrete and Continuous Control Devices”. In: *2018 Power Systems Computation Conference (PSCC)* (June 2018), pp. 1–7. DOI: 10.23919/PSCC.2018.8442555.

[4] Nawaf Nazir, Pavan Racherla, and Mads Almassalkhi. “Optimal multi-period dispatch of distributed energy resources in unbalanced distribution feeders”. In: *arXiv* (June 2019). DOI: 10.48550/arXiv.1906.04108. eprint: 1906.04108.

[5] Xiaodong Qian and Yuanguo Zhu. “Differential Dynamic Programming for Multistage Uncertain Optimal Control”. In: *2014 Seventh International Joint Conference on Computational Sciences and Optimization*. IEEE, July 2014, pp. 88–92. DOI: 10.1109/CSO.2014.25.

APPENDIX

Appendix A

Branch Flow Model: Relaxations and Convexification

In [2] the authors came up the Relaxed Branch Flow Model, and showed that in the case of Tree/Radial networks, the Relaxed Model can solve for the unique optimal solution, including the bus angles, and in the case of weakly meshed networks, there is a mechanism for extracting the bus angles from the relaxed solution, to find out its unique solution, if it exists.

Legend for Table A.2:

Table A.1 Table describing the variables involved in the Branch Flow Model equations.

Symbol	Meaning
p_j, q_j	Real, Reactive Power flowing from bus j into the network.
P_{ij}, Q_{ij}	Real, Reactive Power flowing in branch $i \to j$ (sending-end).
I_{ij}, l_{ij}	Complex Current flowing in branch

Table A.2 Table describing the Branch Flow Model equations.

Equation #	Equation	Unknowns	Knowns	No. of Equations
13	$p_j = \Sigma P_{jk} + \Sigma(P_{ij} - r_{ij}l_{ij}) + g_j v_j$	$1 \times p_0$ $m \times P_{ij}$ $m \times l_{ij}$ $n \times v_j$	$n \times p_j$ $m \times r_{ij}$ $(n+1) \times g_j$ $1 \times v_0$	$(n+1)$
14	$q_j = \Sigma Q_{jk} + \Sigma(Q_{ij} - x_{ij}l_{ij}) + b_j v_j$	$1 \times q_0$ $m \times Q_{ij}$ $m \times l_{ij}$ $n \times v_j$	$n \times q_j$ $m \times x_{ij}$ $(n+1) \times b_j$ $1 \times v_0$	$(n+1)$
15	$v_j = v_i + (r_{ij}^2 + x_{ij}^2)l_{ij} - 2(r_{ij}P_{ij} + x_{ij}Q_{ij})$	$m \times P_{ij}$ $m \times Q_{ij}$ $m \times l_{ij}$ $n \times v_j$	$b \times r_{ij}$ $m \times x_{ij}$ $1 \times v_0$	m
16	$l_{ij} = \frac{P_{ij}^2 + Q_{ij}^2}{v_j}$	$m \times P_{ij}$ $m \times Q_{ij}$ $m \times l_{ij}$ $n \times v_j$	$1 \times v_0$	m
13 to 16		$1 \times p_0$ $1 \times q_0$ $m \times P_{ij}$ $m \times Q_{ij}$ $m \times l_{ij}$ $n \times v_j$	$n \times p_j$ $n \times q_j$ $m \times r_{ij}$ $m \times x_{ij}$ $(n+1) \times g_j$ $(n+1) \times b_j$ $1 \times v_0$	$2(n+1+m)$
		$2(n+1+m)$	$4n+2m+3$	$2(n+1+m)$

Appendix B

Abstracts: Optimization-based Methods for solving MP-OPF

In [4], the authors use a two-step paradigm for solving the MP-OPF problem, by first solving for a more relaxed SOCP problem for the all of the time-steps in a horizon, and using the SOC values from its solution, solve for the NLP OPF problem for every time-step in parallel.

In [3], they prove that for 'realistic' systems, appending an additional 'complementarity' cost function to the original objective function, Simultaneous Charging and Discharging (SCD) in the optimal solution is avoided, and that the Mixed-Integer SOCP problem of AC-OPF with energy storage can be relaxed into a regular SOCP problem without violating the battery physics in the optimal solution.

Appendix C

Abstracts: Dynamic Programming Methods for solving MP-OPF

In [1], the authors use a Differential Dynamic Programming approach, which involved usage of Forward and Backward passes made over a sequence of time-steps, doing a back-and-forth between computation for one time-step, say t, by making some assumptions on any variables required from the next time-step $t+1$, and then updating the assumed values at t, once new values for the $t+1$ time-step have been made.

Appendix D

Abstracts: Differential Dynamic Programming

In [5], the authors lay out the framework on how Differential Dynamic Programming can be utilized to solve a Mutli-Stage Uncertain Optimal Control problem programmatically.

$$
\begin{aligned}
J(x_0,0) = &\min_{\substack{u_j \in U_j \\ j=0,1\ldots N-1}} \mathbb{E}\left[\sum_{j=0}^{N-1}\{L(x_j,u_j,j)\} + G(x_N,N)\right] \\
&s.t. \\
&x_{k+1} = f(x_j,u_j,j) + \sigma_{j+1}\xi_{j+1} \\
&\quad j = 0,1,\ldots N-1
\end{aligned}
\tag{D.1}
$$

$$
J(x_N,N) = G(x_N,N) \tag{D.2}
$$

$$
J(x_k,k) = \min_{u_k \in U_k} \mathbb{E}\left[J(x_{k+1},k+1) + L(x_k,u_k,k)\right] \tag{D.3}
$$

$$
\begin{aligned}
\bar{x}_{k+1} &= f(\bar{x}_k,\bar{u}_k,k) \\
\bar{x}_0 &= x_0
\end{aligned}
\tag{D.4}
$$

$$
\begin{aligned}
f(x_k,u_k,k) = f(\bar{x}_k,\bar{u}_k,k) &+ f_x\delta x_k + f_u\delta u_k \\
&+ \frac{1}{2}f_{xx}(\delta x_k)^2 + \frac{1}{2}f_{uu}(\delta u_k)^2 \\
where& \\
\delta x_k &= x_k - \bar{x}_k \\
\delta u_k &= u_k - \bar{u}_k
\end{aligned}
\tag{D.5}
$$

$$
\begin{aligned}
\delta x_{k+1} &= f(x_k, u_k, k) + \sigma_{k+1}\xi_{k+1} - f(\bar{x}_k, \bar{u}_k, k) \\
\text{or, } \delta x_{k+1} &= f_x \delta x_k + f_u \delta u_k + \frac{1}{2} f_{xu}(\delta x_k)(\delta u_k) \qquad \text{(D.6)} \\
&\quad + \frac{1}{2} f_{xx}(\delta x_k)^2 + \frac{1}{2} f_{uu}(\delta u_k)^2 + \sigma_{k+1}\xi_{k+1}
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
L(x_k, u_k, k) &= L(\bar{x}_k, \bar{u}_k, k) + L_x \delta x_k + L_u \delta u_k \\
&\quad + \frac{1}{2} L_{xx}(\delta x_k)^2 + \frac{1}{2} L_{uu}(\delta u_k)^2 \qquad \text{(D.7)}
\end{aligned}
$$

and,

$$
J(x_k, k) = J(\bar{x}_k, k) + J_x \delta x_k + \frac{1}{2} J_{xx}(\delta x_{k+1})^2 \qquad \text{(D.8)}
$$

Appendix E

Models: Battery Model for Multi-Period OPF

All subscripts j for a variable imply the node in the power grid (for node j). All superscripts t refer to the time period number t.

In [3, 4], the batteries are modelled using four state/control variables, which are:

TABLE E.1 Description of Grid Parameters

Variable	Description
$\mathcal{N}$	Set of all nodes. $\mathcal{N} = \{1, 2, \ldots n\}$
$\mathcal{L}$	Set of all branches. $\mathcal{L} = \{1, 2, \ldots l\} = \{(i, k)\} \subset (\mathcal{N} \times \mathcal{N})$.
$\mathcal{D}$	Set of all nodes containing DERs. $\mathcal{D} \subset \mathcal{N}$
$\mathcal{B}$	Set of all nodes containing storage. $\mathcal{B} \subset \mathcal{N}$
Δt	Duration of a single time period. Here $\Delta t = 15$ min. $= 0.25$ h.
T	Prediction Horizon Duration. Total time duration solved for as part of one instance of MP-OPF.
K	Prediction Horizon Number. Total number of discrete-time steps solved for in one instance of MP-OPF. $T = K\Delta t$

TABLE E.2 Description of Branch Flow Model Variables

Variable	Description
p_j^t	Fixed Real Power Generation minus Fixed Real Power Load. Here, $p_j^t = {p_{Dj}}^t - {p_{Lj}}^t$. A known (predicted) value $\forall\, t, j$.
q_j^t	Fixed Reactive Power Generation minus Fixed Reactive Power Load. Here, $q_j^t = -{q_{Lj}}^t$. A known (predicted) value $\forall\, t, j$.
$p_{D_j}^t$	Real Power Generated by DERs
$p_{L_j}^t$	Real Power Demand
$q_{L_j}^t$	Reactive Power Demand

TABLE E.3 Description of Battery Variables

Variable	Description	Dimension	Dimension pu
$B_{n,k}$	State of Charge (SOC) of Battery after time-interval k	$[kWh]$	$[pu\,h]$
$P^c_{n,k}$	Average Charging Power of the Battery during the k-th time interval.	$[kW]$	$[pu]$
$P^d_{n,k}$	Average Discharging Power of the Battery during the k-th time interval.	$[kW]$	$[pu]$
$q_{B_{n,k}}$	Average Reactive Power Output from the Battery Inverter during the k-th time interval.	$[kVAr]$	$[pu]$

where,

$k \in \{0, 1, 2, \ldots, T/\Delta t\}$ The index of the discretized time intervals, where k represents the k-th time interval of duration Δt within the continuous time range $[0, T]$. The corresponding continuous time interval is $[(k-1)\Delta t, k\Delta t]$.

$n \in \mathcal{N}$ The node n is an element of the set of all nodes in the power grid $\mathcal{N}$. Note that n can be used both as an iterator or as the total number of nodes in the grid (i.e. the cardinality of $\mathcal{N}$), and its meaning should be obvious from context.

Table E.4 Values, Lower Bounds and Upper Bounds on Battery Variables

Variable	Value or Limits	Description
P_{Max}	P_{Rated} of corresponding DER.	
$q_{B_{Max}}$	$q_{D_{Rated}}$ of corresponding DER.	
P_d, P_c	$[0, P_{Max}]$	
E_{Rated}	$P_{Max} \times 4$ h	4 h of one-way Charging/Discharging at Maximum Power
B	$[0.30E_{Rated}, 0.95E_{Rated}]$	2.4 h of one-way Charging/Discharging at Maximum Power
B_0	$0.625E_{Rated}$	Batteries start with an SOC value in the middle of their SOC range.
η_d, η_c	0.95	
α	3e−5	Value depends on the magnitude of the loss term in the objective function (for IEEE 123 node system, it is $\approx (1\text{e}{-5}, 10\text{e}{-5})$). Too big a value of α would reduce both P_c and P_d terms to zero, whereas too small a value would not penalize SCD, causing physically infeasible solutions.

Optimization Equations

Original Problem: Mixed-Integer Nonlinear Optimization Model - Full Horizon

$$\min_{P_{ij}^t, Q_{ij}^t, v_j^t, l_{ij}^t, q_{D_j}^t, B_j^t, P_{c_j}^t, {P_{d_j}}^t, q_{B_j}^t} \quad \sum_{t=1}^{T} \sum_{i \to j \in \mathcal{L}} (r_{ij} l_{ij}^t) \tag{E.1}$$

s.t.

$$p_j^t = \sum_{j \to k \in \mathcal{L}} P_{jk}^t - \sum_{i \to j \in \mathcal{L}} \left\{P_{ij}^t - r_{ij} l_{ij}^t\right\} - P_{d_j}^t + P_{c_j}^t \tag{E.2}$$

$$q_j^t = \sum_{j \to k \in \mathcal{L}} Q_{jk}^t - \sum_{i \to j \in \mathcal{L}} \left\{Q_{ij}^t - x_{ij} l_{ij}^t\right\} - q_{D_j}^t - q_{B_j}^t \tag{E.3}$$

$$v_j^t = v_i^t + \left\{r_{ij}^2 + x_{ij}^2\right\} l_{ij}^t - 2(r_{ij} P_{ij}^t + x_{ij} Q_{ij}^t) \tag{E.4}$$

$$l_{ij}^t = \frac{(P_{ij}^t)^2 + (Q_{ij}^t)^2}{v_i^t} \tag{E.5}$$

$$B_j^t = B_j^{t-1} + z \Delta t \eta_c P_{c_j}^t - (1-z) \Delta t \frac{1}{\eta_d} P_{d_j}^t \tag{E.6}$$

$$B_j^0 = 0.5(soc_{max} + soc_{min}) E_{Rated} = 0.625 E_{Rated} \tag{E.7}$$

$$where, \tag{E.8}$$

$$p_j^t = {p_D}_j^t - {p_L}_j^t \tag{E.9}$$

$$q_j^t = -{q_L}_j^t \tag{E.10}$$

$$t = \{1, 2, \ldots T\} \tag{E.11}$$

$$z = \{0, 1\} \tag{E.12}$$

(Integer Constraint Relaxed) Naive Brute Force Full Optimization Model - Full Horizon

$$\min_{P_{ij}^t, Q_{ij}^t, v_j^t, l_{ij}^t, q_{D_j}^t, B_j^t, P_{c_j}^t, {P_{d_j}}^t, q_{B_j}^t} \quad \sum_{t=1}^{T} \sum_{i \to j \in \mathcal{L}} (r_{ij} l_{ij}^t) + \alpha \sum_{t=1}^{T} \sum_{j \in \mathcal{B}} \left\{ (1-\eta_c) P_{c_j}^t + \left(\frac{1}{\eta_d} - 1 \right) P_{d_j}^t \right\} \tag{E.13}$$

s.t.

$$p_j^t = \sum_{j \to k \in \mathcal{L}} P_{jk}^t - \sum_{i \to j \in \mathcal{L}} \left\{ P_{ij}^t - r_{ij} l_{ij}^t \right\} - P_{d_j}^t + P_{c_j}^t \tag{E.14}$$

$$q_j^t = \sum_{j \to k \in \mathcal{L}} Q_{jk}^t - \sum_{i \to j \in \mathcal{L}} \left\{ Q_{ij}^t - x_{ij} l_{ij}^t \right\} - q_{D_j}^t - q_{B_j}^t \tag{E.15}$$

$$v_j^t = v_i^t + \left\{ r_{ij}^2 + x_{ij}^2 \right\} l_{ij}^t - 2(r_{ij} P_{ij}^t + x_{ij} Q_{ij}^t) \tag{E.16}$$

$$l_{ij}^t = \frac{(P_{ij}^t)^2 + (Q_{ij}^t)^2}{v_i^t} \tag{E.17}$$

$$B_j^t = B_j^{t-1} + \Delta t \eta_c P_{c_j}^t - \Delta t \frac{1}{\eta_d} P_{d_j}^t \tag{E.18}$$

$$B_j^0 = 0.5(soc_{max} + soc_{min}) E_{Rated} = 0.625 E_{Rated} \tag{E.19}$$

$$where, \tag{E.20}$$

$$p_j^t = {p_{D_j}}^t - {p_{L_j}}^t \tag{E.21}$$

$$q_j^t = -{q_{L_j}}^t \tag{E.22}$$

$$t = \{1, 2, \ldots T\} \tag{E.23}$$

Step 2: Full Optimization Model - Single Time Step Greedy Approach

$$\min_{P^t_{ij},Q^t_{ij},v^t_j,l^t_{ij},q^t_{D_j},B^t_j,P^t_{c_j},P_{dj}{}^t,q^t_{B_j}} \quad \sum_{i\to j\in\mathcal{L}}(r_{ij}l^t_{ij}) + \alpha\sum_{j\in\mathcal{B}}\left\{(1-\eta_c)P^t_{c_j} + \left(\frac{1}{\eta_d}-1\right)P^t_{d_j}\right\} \tag{E.24}$$

s.t.

$$p^t_j = \sum_{j\to k\in\mathcal{L}} P^t_{jk} - \sum_{i\to j\in\mathcal{L}}\left\{P^t_{ij} - r_{ij}l^t_{ij}\right\} - P^t_{d_j} + P^t_{c_j} \tag{E.25}$$

$$q^t_j = \sum_{j\to k\in\mathcal{L}} Q^t_{jk} - \sum_{i\to j\in\mathcal{L}}\left\{Q^t_{ij} - x_{ij}l^t_{ij}\right\} - q^t_{D_j} - q^t_{B_j} \tag{E.26}$$

$$v^t_j = v^t_i + \left\{r^2_{ij} + x^2_{ij}\right\}l^t_{ij} - 2(r_{ij}P^t_{ij} + x_{ij}Q^t_{ij}) \tag{E.27}$$

$$l^t_{ij} = \frac{(P^t_{ij})^2 + (Q^t_{ij})^2}{v^t_i} \tag{E.28}$$

$$B^t_j = B^{t-1}_j + \Delta t\eta_c P^t_{c_j} - \Delta t\frac{1}{\eta_d}P^t_{d_j} \tag{E.29}$$

$$B^0_j = 0.5(soc_{max} + soc_{min})E_{Rated} = 0.625E_{Rated} \tag{E.30}$$

$$where, \tag{E.31}$$

$$p^t_j = p_{Dj}{}^t - p_{Lj}{}^t \tag{E.32}$$

$$q^t_j = -q_{Lj}{}^t \tag{E.33}$$

$$t = \{1, 2, \dots T\} \tag{E.34}$$

Step 1b: Initialisation Lossless Optimization Model WITH Batteries - Single Time Step Greedy Approach

$$\min_{P_{ij}^t, Q_{ij}^t, q_{D_j}^t, B_j^t, P_{c_j}^t, P_{d_j}{}^t, q_{B_j}^t} \quad \sum_{i \to j \in \mathcal{L}} \left\{ r_{ij} \frac{(P_{ij}^t)^2 + (Q_{ij}^t)^2}{v_{0i}^t} \right\} + \alpha \sum_{j \in \mathcal{B}} \left\{ (1 - \eta_c) P_{c_j}^t + \left(\frac{1}{\eta_d} - 1 \right) P_{d_j}^t \right\} \tag{E.35}$$

s.t.

$$p_j^t = \sum_{j \to k \in \mathcal{L}} P_{jk}^t - \sum_{i \to j \in \mathcal{L}} \left(P_{ij}^t \right) - P_{d_j}^t + P_{c_j}^t \tag{E.36}$$

$$q_j^t = \sum_{j \to k \in \mathcal{L}} Q_{jk}^t - \sum_{i \to j \in \mathcal{L}} \left(Q_{ij}^t \right) - q_{D_j}^t - q_{B_j}^t \tag{E.37}$$

$$v_{0j}^t = v_{0i}^t - 2(r_{ij} P_{ij}^t + x_{ij} Q_{ij}^t) \tag{E.38}$$

$$B_j^t = B_j^{t-1} + \Delta t \eta_c P_{c_j}^t - \Delta t \frac{1}{\eta_d} P_{d_j}^t \tag{E.39}$$

$$B_j^0 = 0.5(soc_{max} + soc_{min}) E_{Rated} = 0.625 E_{Rated} \tag{E.40}$$

$$where, \tag{E.41}$$

$$p_j^t = p_{Dj}{}^t - p_{Lj}{}^t \tag{E.42}$$

$$q_j^t = -q_{Lj}{}^t \tag{E.43}$$

$$t = \{1, 2, \dots T\} \tag{E.44}$$

Step 1a: Initialisation Lossless Optimization Model WITHOUT Batteries - Single Time Step Greedy Approach

$$\min_{P_{ij}^t, Q_{ij}^t, v_j^t, q_{D_j}^t} \quad 0 \tag{E.45}$$

s.t.

$$p_j^t = \sum_{j \to k \in \mathcal{L}} P_{jk}^t - \sum_{i \to j \in \mathcal{L}} P_{ij}^t \tag{E.46}$$

$$q_j^t = \sum_{j \to k \in \mathcal{L}} Q_{jk}^t - \sum_{i \to j \in \mathcal{L}} Q_{ij}^t \tag{E.47}$$

$$v_j^t = v_i^t - 2(r_{ij} P_{ij}^t + x_{ij} Q_{ij}^t) \tag{E.48}$$

$$where, \tag{E.49}$$

$$p_j^t = {p_{Dj}}^t - {p_{Lj}}^t \tag{E.50}$$

$$q_j^t = -{q_{Lj}}^t \tag{E.51}$$

$$t = \{1, 2, \ldots T\} \tag{E.52}$$

A simple metric called P_{Save} gives an indication of the effect of power generated by batteries and DERs which offset substation power (indirectly flowing it via its parent area if it is not directly connected to the substation).

Its formula is as shown:

$$P_{Save} = 100\% * \left(\frac{\sum_{j \in \mathcal{B}} \left(P_{d_j} - P_{c_j}\right)}{P_{12} + \sum_{j \in \mathcal{D}} P_{DER_j} + \sum_{j \in \mathcal{B}} \left(P_{d_j} - P_{c_j}\right)} \right) \tag{E.53}$$