Random Batch Method (RBM) and Model Predictive Control (MPC) for Converter-Dominated Power Systems

Carlos Vázquez Monzón

July 29, 2024

Abstract

This paper investigates the application of Data-based Reduced Order Models, the Random Batch Method (RBM) and Model Predictive Control (MPC) in Converter-Dominated Power Systems. The study evaluates their effectiveness in handling stiff systems, marking a pioneering effort in this domain. Several challenges emerge, particularly concerning the accuracy of approximations due to the stiffness of the state matrix A: block-wise decomposition strategies face limitations in accurately approximating the original, as evidenced by simulation results. This limitation questions the effectiveness of block-wise strategies, highlighting the need for alternative decomposition methods. Spectral decomposition showed potential but prompts a search for more efficient alternatives. Also, we show that combining MPC with RBM is a promising approach for optimal control problems constrained by state trajectories, as the latter efficiently handles large time horizons T by breaking down the time into smaller sub-intervals, and although a marginal improvement of the RBM-MPC over RBM was found, we have concluded that does not justify the increased computation time, emphasizing the need to understand factors influencing error accumulation. Furthermore, the study identifies a relationship between the RBM, the number of discretization points, and the switching parameter K, where increasing these parameters enhances accuracy but may lead to deterioration in quality. These results are in line with the theoretical framework. Overall, the study highlights the necessity for continued investigation to optimize control strategies for stiff systems such as power systems.

Introduction

The integration of renewable energy sources and power electronic converters into modern power systems has brought forth a myriad of challenges associated with system dynamics. Effectively managing these complexities is imperative to ensure the stability and reliability of Converter-Dominated Power Systems.

Data-based Reduced Order Models ([9], [8], [3]) offer a promising approach by simplifying the complex system dynamics into more manageable forms without compromising essential characteristics. In the context of Converter-Dominated Power Systems, the RBM stands out for its ability to handle large-scale problems by decomposing them into smaller, more tractable subproblems ([2], [4], [6], [6], [7], [11]). Unlike traditional deterministic methods, RBM embraces the stochastic nature of renewable energy sources and system uncertainties, offering a more realistic depiction of the dynamic behavior of the power system.

The fundamental concept underlying the RBM involves the creation of random matrices by gener-

ating batches representing the indices of a previous decomposition of the state matrix of the system (see Section 4.1). This only can be done, presumably, in the context of linear optimal control. and due to the non-linearity of the considered model, we need to linearize it (Section 2).

The primary focus of this study is to evaluate the effectiveness of these methodologies in handling stiff systems—a critical aspect that has not been extensively explored in the literature. Stiff systems, characterized by widely varying timescales in their dynamics, present significant challenges for numerical approximation and control . The stiffness of the state matrix A in such systems complicates accurate approximation, particularly when using block-wise decomposition strategies. This research highlights the limitations of these traditional strategies and explores alternative approaches, including spectral decomposition, to improve accuracy.

This paper aims to explore as well the application of the RBM in tandem with optimal control techniques, such as the MPC, for Converter-Dominated Power Systems. The combined approach aims to capitalize on the strengths of both methodologies, addressing found challenges while optimizing control actions to enhance system performance. However, the study also uncovers the trade-offs involved, such as increased computational time versus marginal improvements in accuracy.

To implement the RBM and optimal control strategies, we turn to Python, a versatile programming language with extensive libraries and tools for scientific computing and control system analysis. Python facilitates the implementation of complex algorithms, making it an ideal choice for exploring the synergies between RBM and optimal control in the context of power systems. In the upcoming sections, we will delve into the technical aspects of implementing RBM and MPC using Python, providing a practical guide for researchers and practitioners in the field.

1 Converter-Dominated Power Systems

A Converter-Dominated Power System typically involves the integration of renewable energy sources, such as wind and solar, through power electronic converters. These converters play a crucial role in converting the variable and intermittent output from renewable sources into a form suitable for the power grid. Additionally, grid-forming converters may be employed to actively control the voltage and frequency of the system.

These power systems typically include three elements:

- A power *grid*. Power grids are networks of interconnected components that work together to generate, transmit, and distribute electrical energy.

- A *converter*, particularly an *inverter*. Converters are devices that transform electrical energy from one form to another, often involving the conversion between alternating current (AC) and direct current (DC). The inverters convert DC from sources like solar panels or batteries into AC for grid integration. In our case, we consider a *synchronverter*, which is a type of inverter that mimics synchronous generators, which offers a mechanism for power systems to control grid-connected renewable energy and facilitates smart grid integration.

- A *controller*. Due to the very nature of the synchronverter, it provides an outer-loop controller [13]. Therefore, we can control the amplitude, frequency, and phase angle of the generated voltage, allowing for precise adjustment to match grid requirements.

2 Mathematical Model

We consider only one synchronverter (1 converter + 1 controller). With that, the model is the following ODE system:

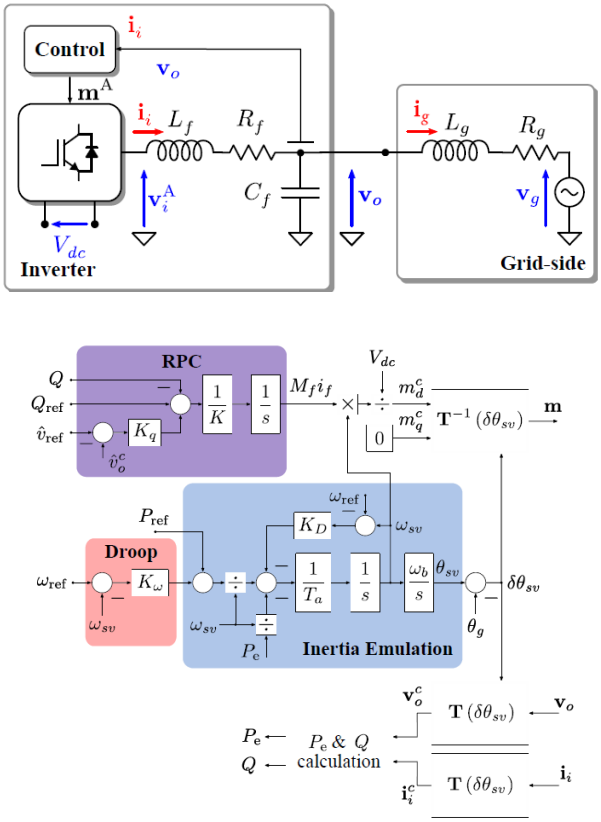

Figure 1: Diagram of the considered power system (adapted from [1])

.

$$
\begin{cases}
\frac{di_g}{dt} & = \frac{1}{L_g}(v_0 - R_g i_g - v_g) - j\omega_g i_g \\
\frac{di_i}{dt} & = \frac{1}{L_f}(\mathbf{m}V_{dc} - R_f i_i - v_0) - j\omega_g i_i \\
\frac{dv_0}{dt} & = \frac{1}{C_f}(i_i - i_g) - j\omega_g v_0 \\
\frac{d(M_f i_f)}{dt} & = \frac{1}{K}\left[Q_{ref} - Q + K_q(\hat{v}_{ref} - \hat{v}_0^c)\right] \\
\frac{d\omega_{sv}}{dt} & = \frac{1}{T_a}\left[\frac{P_m}{\omega_{sv}} - \frac{P_e}{\omega_{sv}} - K_D(\omega_{sv} - \omega_{ref})\right] \\
\frac{d\delta\theta_{sv}}{dt} & = \omega_{sv} - \omega_g,
\end{cases}
\tag{1}
$$

where

$$
\begin{aligned}
P_m &= P_{ref} + K_w(\omega_{ref} - \omega_{sv}), \\
\mathbf{v}_{o_c} &= \begin{bmatrix} \cos(\delta\theta_{sv}) & \sin(\delta\theta_{sv}) \\ -\sin(\delta\theta_{sv}) & \cos(\delta\theta_{sv}) \end{bmatrix} \begin{bmatrix} v_0^d \\ v_0^q \end{bmatrix}, \\
\hat{v}_o^c &= \sqrt{v_{o_c,0}^2 + v_{o_c,1}^2}, \\
\mathbf{i}_{i_c} &= \begin{bmatrix} \cos(\delta\theta_{sv}) & \sin(\delta\theta_{sv}) \\ -\sin(\delta\theta_{sv}) & \cos(\delta\theta_{sv}) \end{bmatrix} \begin{bmatrix} i_i^d \\ i_i^q \end{bmatrix}, \\
P_e &= v_{o_c,0} i_{i_c,0} + v_{o_c,1} i_{i_c,1}, \\
Q &= -v_{o_c,0} i_{i_c,1} + v_{o_c,1} i_{i_c,0}, \\
\mathbf{m} &= \begin{bmatrix} \cos(\delta\theta_{sv}) & -\sin(\delta\theta_{sv}) \\ \sin(\delta\theta_{sv}) & \cos(\delta\theta_{sv}) \end{bmatrix} \begin{bmatrix} \frac{\omega_{sv} \cdot M_f i_f}{V_{dc}} \\ 0 \end{bmatrix}, \\
m_d &= m_{,0}, \\
m_q &= m_{,1}.
\end{aligned}
$$

j is the imaginary unit. v_0, i_g, v_g, m and i_i have two components: one in the d-axis (real part) and the other in the q-axis (imaginary part). We can therefore convert the complex ODE system into a real one, by introducing the variables $v_0^q, v_0^d, i_g^d, i_g^q$, etc. The variables highlighted in blue, orange and red represent the states, the controls and the parameters, respectively.

Physically, the variables represent:

L_f, C_f, R_f - Inductance, capacitance and resistance of the inverter, respectively.

V_{dc} - DC voltage

v_0 - Output voltage

L_g, C_g, R_g - Inductance, capacitance and resistance of the grid, respectively.

i_g - grid current

v_g - grid voltage

ω_{sv} - controller frequency

m - modulation index from the controller

i_i - input current of the inverter

$M_f i_f$ - reactive power of the controller

$\delta\theta_{sv}$ - angle between the grid and the controller reference plane

$Q, Q_{ref}, \hat{v}_{ref}, K$ and K_{ref} - variables governing the RPC part of the synchronverter (voltage regulator).

ω_{ref}, K_ω - variables governing the droop of the synchronverter (frequency regulator).

P_{ref}, K_D, T_a - variables governing the inertia simulator of the synchronverter.

2.1 Linearized model

In order to combine the framework of Linear-Quadratic Optimal control with our RBM methodology, we need to define our state matrix A. Due to the non-linearity of the model defined in (1), our next step was to linearize the system.

We can find a steady-state point x_0 of (1) for u_0 = ($398\sqrt{2}$, 0, 1375000, 0, 100π, 563). We obtain the linearized model by evaluating the gradient with respect to the controls u and the states x in x_0:

$$\begin{aligned} x' = f(x,u) \approx & \nabla_x f(x_0,u_0)x + \nabla_u f(x_0,u_0)u \\ & Ax + Bu, \quad x(0) = x_0, \end{aligned}$$

where f is defined as the right-hand side of (1). As initial condition, we also choose x_0.

The state and control matrices A and B are illustrated in the Appendix A. Note that the size of A and B is 9x9 and 9x6, respectively.

3 Optimal Control Problem

We consider the classical (finite-dimensional) LQ optimal control problem in which we want to find the control $u^*(t)$ that minimizes

$$\begin{aligned} J(\mathbf{x}(t),\mathbf{u}(t)) = & \frac{1}{2}\int_0^T (x(t)-x_d(t))^T Q(x(t)-x_d(t)) + \\ & \frac{1}{2}\int_0^T u(t)^T R u(t), \end{aligned} \tag{2}$$

subject to the dynamics

$$\dot{x}(t) = Ax(t) + Bu(t),\ x(0) = x_0, \tag{3}$$

with x_d being the reference trajectory and Q and R the weighting matrices. In our case, we set Q and R as the identity matrix.

In this context, $x(t)$ and $x_d(t)$ evolve in $\mathbb{R}^N$, while $u(t)$ evolves in $\mathbb{R}^q$. For scenarios with a large N, conventional problem-solving involves gradient-based techniques like gradient descent or conjugate gradients. The gradients required for these methods are efficiently computed as

$$(\nabla J(u))(t) = B^\top \varphi(t) + Ru(t),$$

where the adjoint state $\varphi(t)$ adheres to

$$-\dot{\varphi}(t) = A^\top \varphi(t) + Q(x(t) - x_d(t)), \qquad \varphi(T) = 0.$$

It's important to note that determining $\varphi(t)$ involves integrating backward in time from the final condition $\varphi(T) = 0$. Additionally, $x(t)$ in the ordinary differential equation for $\varphi(t)$ represents the solution of the state equation, where $u(t)$ signifies the point at which the gradient is computed. This

implies that both the forward dynamics for $x(t)$ and the backward dynamics for $\varphi(t)$ must be solved in each iteration.

Practical implementation necessitates the discretization of the aforementioned equations in time. This approach becomes computationally demanding when A lacks sparsity and is of significant size, coupled with a large number of points in the time grid. These computational challenges will be discussed in more detail below.

The subsequent sections will delve into the specifics of the RBM methodology, optimal control strategies, and their combined application to address the challenges in power system control.

4 Random Batch Method

The RBM is a numerical approach used in the context of optimization and sampling for solving various problems, including optimization problems, machine learning, and computational mathematics. It involves using random batches or subsets of data to approximate quantities, gradients, or solutions to problems. The method has been particularly applied in the field of optimization, where it can be employed to approximate the objective function, gradients, or Hessian matrix using randomly selected subsets of the data ([7], [5], [4], [11]).

4.1 The proposed randomized time-splitting method

In order to implement the RBM, we need to split the matrix following the method of [11] and [10].

Decomposition of Matrix A

Decompose the matrix A into M submatrices A_m:

$$A = \sum_{m=1}^{M} A_m \tag{4}$$

Assignment of Probabilities

For each of the 2^M subsets of $\{1, 2, ..., M\}$, denoted as $\{S_1, S_2, ..., S_{2^M}\}$, assign probabilities $p_1, p_2, ..., p_{2^M}$ such that:

$$\sum_{l=1}^{2^M} p_l = 1 \tag{5}$$

and

$$\pi_m = \sum_{l \in L_m} p_l > 0, \tag{6}$$
$$L_m = \{l \in \{1, 2, ..., 2^M\} / m \in S_l\},$$

for all $m \in \{1, 2, ..., M\}$.

Time Interval Division

Divide the considered time interval $[0, T]$ into N sub-intervals $[t_{n-1}, t_n)$, $n \in \{1, ..., N\}$:

$$0 = t_0 < t_1 < ... < t_{N-1} < t_N = T \tag{7}$$

and choose an index ω_n according to the probability distribution of the assigned probabilities $p_1, p_2, ..., p_{2^M}$ in each sub-interval independently. Store the selected indices as

$$\omega := (\omega_1, \omega_2, ..., \omega_N). \tag{8}$$

Matrix Definition

For the selected ω, define a matrix $A_h(\omega, t)$:

$$A_h(\omega, t) = \sum_{m \in S_{\omega_k}} \frac{A_m}{\pi_m} \tag{9}$$

for $t \in [t_{n-1}, t_n), n \in \{1, ..., N\}$. It can be easily proven that $\mathbb{E}(A_h(\omega, t)) = A$ [11].

Solution Computation

Replace the matrix A by $A_h(\omega, t)$ in Equations (1) and (2). Now compute the solution $x_h(\omega, t)$ of the dynamics:

$$\dot{x}_h(\omega, t) = A_h(\omega, t) x_h(\omega, t) + Bu(t), \; x_h(\omega, 0) = x_0 \tag{10}$$

for a given control $u(t)$.

4.2 The batch

From the very definition of $A_h(\omega, t)$, we can see that it is piece-wise constant, since ω is fixed. Then, it can take exactly N different values, being N the number of timestep. So, we can define a vector with N random components, where each component is a random number between 0 and $M-1$. That is the so-called *batch*. We can, therefore, select one value of $A_h(\omega, t)$ in each timestep by selecting the component of the batch associated with that timestep.

Shift parameter of the batch K

As in [11], we can study the RBM by constructing different methods of generating the batch. One way to do this is by choosing the period of the randomization of the batch components. For example, if we choose $K = 2$, the batch changes values each 2 indices.

5 Technicalities of the electrical model

The initial idea was to split the 9 x 9 matrix A in 3 x 3 blocks:

$$A = \begin{pmatrix} B_1 & B_2 & B_3 \\ B_4 & B_5 & B_6 \\ B_7 & B_8 & B_9 \end{pmatrix},$$

so A $= \sum_{i=1}^{6} A_i$, with

$$A_1 = \begin{pmatrix} B_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, A_2 = \begin{pmatrix} 0 & B_2 & 0 \\ B_4 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$A_3 = \begin{pmatrix} 0 & 0 & B_3 \\ 0 & 0 & 0 \\ B_7 & 0 & 0 \end{pmatrix}, A_4 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & B_5 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$

$$A_5 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & B_6 \\ 0 & B_8 & 0 \end{pmatrix}, A_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & B_9 \end{pmatrix}.$$

With that in mind, the simulations of the RBM for the first state and control follow (Figures 2 and 3).

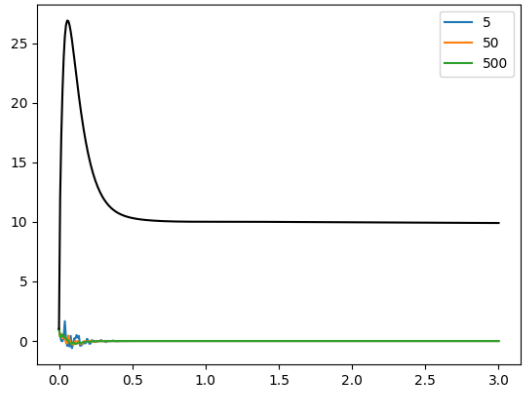

Figure 2: RBM for the state 0 for 5, 50 and 500 trials with the block-wise decomposition

The black lines represent the state/control of the original matrix A. However, the obtained results are unsatisfactory as the approximations do not effectively represent the solution, regardless of the number of trials. This supports the idea that A is not conducive to being split into blocks, as confirmed in Discussions. Because of that, we opt for a *spectral* decomposition.

The eigenvalues of the state matrix A are: $-6.42 \pm 5692.42j, -6.42 \pm 5064.10j, -14.42 \pm 314.16j, -39.44, -23.64$ and $-4.65 \cdot 10^{-3}$. All eigenvalues have real negative part. Note that this is an extremely *stiff* matrix, where $\text{cond}(A) > 10^8$. This is further reinforced by the clear difference between the largest and the smallest eigenvalue module.

5.1 Initial matrix decomposition

Since $\det(A) \neq 0$, we can use the eigenvalue decomposition

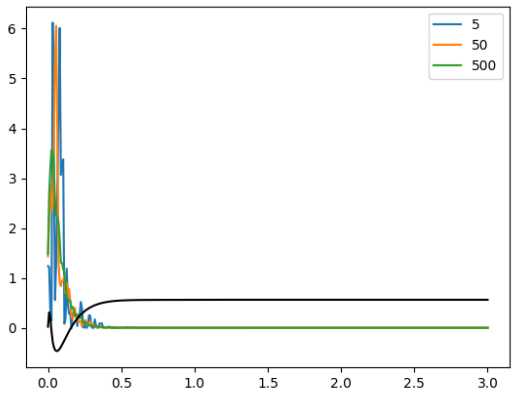

Figure 3: RBM for the control 0 for 5, 50 and 500 trials with the block-wise decomposition

$$A = Q\Lambda Q^{-1}, \ sp(A) = \{\lambda_1, \lambda_2, ..., \lambda_9\}.$$

where Q is the matrix of eigenvectors and Λ is the diagonal matrix of eigenvalues. The spectral decomposition is then formulated as:

$$\Lambda = \Lambda_{s_1} + \Lambda_{s_2} + \Lambda_{s_3},$$

$$\begin{aligned}\Lambda_{s_1} &= diag(\lambda_5, \lambda_9, \lambda_8, 0, 0, ..., 0),\\ \Lambda_{s_2} &= diag(0, 0, 0, \lambda_6, \lambda_7, 0, 0, ..., 0),\\ \Lambda_{s_3} &= diag(0, 0, ..., 0, \lambda_1, \lambda_2, \lambda_3, \lambda_4)\end{aligned}$$

Then, the initial matrix splitting is:

$$A = \sum_{i=1}^{3} A_i = A_1 + A_2 + A_3, \tag{11}$$

where

$$\begin{aligned}A_1 &= Q\Lambda_{s_1}Q^{-1},\\ A_2 &= Q\Lambda_{s_2}Q^{-1},\\ A_3 &= Q\Lambda_{s_3}Q^{-1}.\end{aligned}$$

5.2 Assignment of Probabilities

We assign the same probability to every singleton in the set of all subsets of {1,2,...,M}, and zero to any other subset:

$$p_l = \begin{cases} 1/3 & \text{if } S_l \in \{\{1\}, \{2\}, \{3\}\} \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

6 Algorithm of the RBM

Initialization:

- Define time step differences: $h = \Delta t = t_1 - t_0$.
- Set maximum iterations: max_iters $= 500$.
- Set tolerance for convergence: tol $= 1e - 6$.

Iterations:

For each iteration _ from 1 to max_iters:

- Calculate states $xsol0$ using the function `CalculateStatesRandom` with parameters $A, B, u0, tgrid, X0$, and `batches`.
- Compute the cost function $J0$ using the function `CostFunction` with parameters $Q, R, xd, xsol0, u0, tgrid$, and Δt.
- Compute ϕ using the function `ComputePhiRBM` with parameters $A, Q, x_{sol0}, xd, tgrid$, and `batches`.
- Compute the gradient of the cost function ∇J as gradJ $= B^T\phi + 2Ru0$.
- Compute the inner product G of ∇J with itself over time: $G =$ InnerProduct(gradJ, gradJ, Δt).
- Calculate states dx using the function `CalculateStatesRandom` with parameters $A, B,$ gradJ$, tgrid,$ zeros_like$(X0)$, and `batches`.
- Compute the Hessian matrix H using the function `Hessian` with parameters $Q, R, dx,$ gradJ, and Δt.
- Compute the step size step $= \frac{G}{H}/2$.

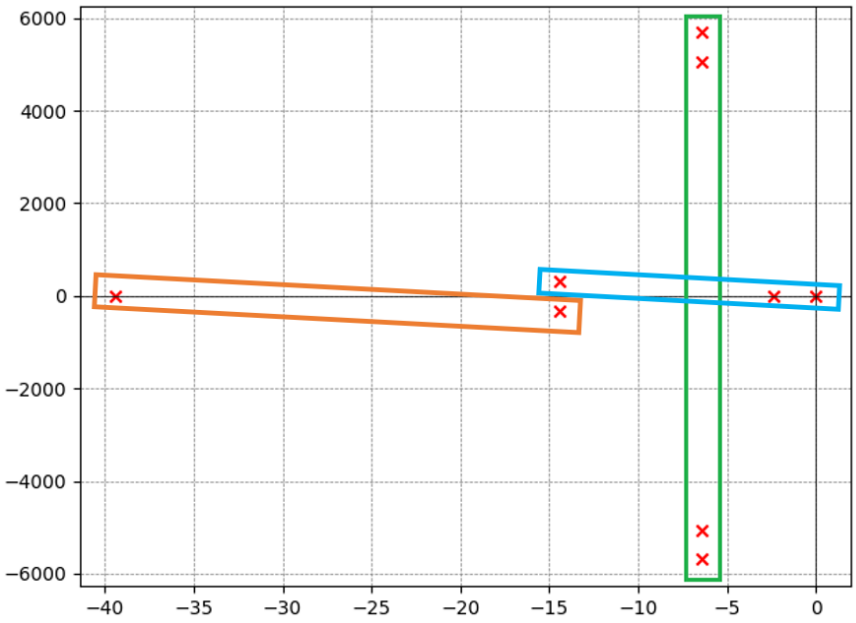

Figure 4: Plot of the eigenvalues, with the 3 sets of eigenvalues associated to the 3 subspaces

- Update the control input $u1$ as $u1 = u0 - \text{step} \cdot \text{gradJ}$.
- Update the states $xsol1$ as $xsol1 = xsol0 - \text{step} \cdot dx$.
- Compute the new cost function $J1$ using the function `CostFunction` with parameters $Q, R, xd, xsol1, u1, tgrid$, and Δt.

Convergence Check:

If $||J1 - J0|| < \text{tol} \cdot |J0|$, return the current control input $u1$ and the initial cost $J0$ as the solution.

Update Control Input:

Update the control input for the next iteration: $u0 = u1$.

Stop criterion

- If the maximum number of iterations is reached without convergence, print 'Not converged' and set the control input $u0$ to None.
- Return the final control input $u0$ and the initial cost $J0$.

7 Model Predictive Control for Power Systems

The Model Predictive Control (MPC) approach is explored as a robust control strategy for Converter-Dominated Power Systems.

- Let $\tau_i := \tau i$, $i \geq 0$, in the subsequent steps.
- Beginning at $i = 0$, we forecast over $[\tau_i, \tau_i + T]$, and obtain an optimal control $\boldsymbol{u}^*(t)$, which

Algorithm 1 Algorithm for the RBM

procedure ComputeControlRBM($A, X0, B, u0, Q, R, xd, tgrid, batches$)
  $dt \leftarrow \text{diff}(tgrid)$
  $max\_iters \leftarrow 500$
  $tol \leftarrow 1e-6$
  for $\_ \leftarrow 1$ **to** max_iters **do**
    $xsol0 \leftarrow \text{CalculateStatesRandom}(A, B, u0, tgrid, X0, batches)$
    $J0 \leftarrow \text{CostFunction}(Q, R, xd, xsol0, u0, tgrid, dt)$
    $\phi \leftarrow \text{ComputePhiRBM}(A, Q, xsol0, xd, tgrid, batches)$
    $gradJ \leftarrow B^T\phi + 2Ru0$
    $G \leftarrow \text{InnerProduct}(gradJ, gradJ, dt)$
    $dx \leftarrow \text{CalculateStatesRandom}(A, B, gradJ, tgrid, \text{zeros\_like}(X0), batches)$
    $H \leftarrow \text{Hessian}(Q, R, dx, gradJ, dt)$
    $step \leftarrow \frac{G}{H}/2$
    $u1 \leftarrow u0 - step \cdot gradJ$
    $xsol1 \leftarrow xsol0 - step \cdot dx$
    $J1 \leftarrow \text{CostFunction}(Q, R, xd, xsol1, u1, tgrid, dt)$
    if $|J1 - J0| < tol \cdot |J0|$ **then**
      return $u1, J0$
    end if
    $u0 \leftarrow u1$
  end for
  print('Not converged')
  $u0 \leftarrow \text{None}$
  return $u0, J0$
end procedure

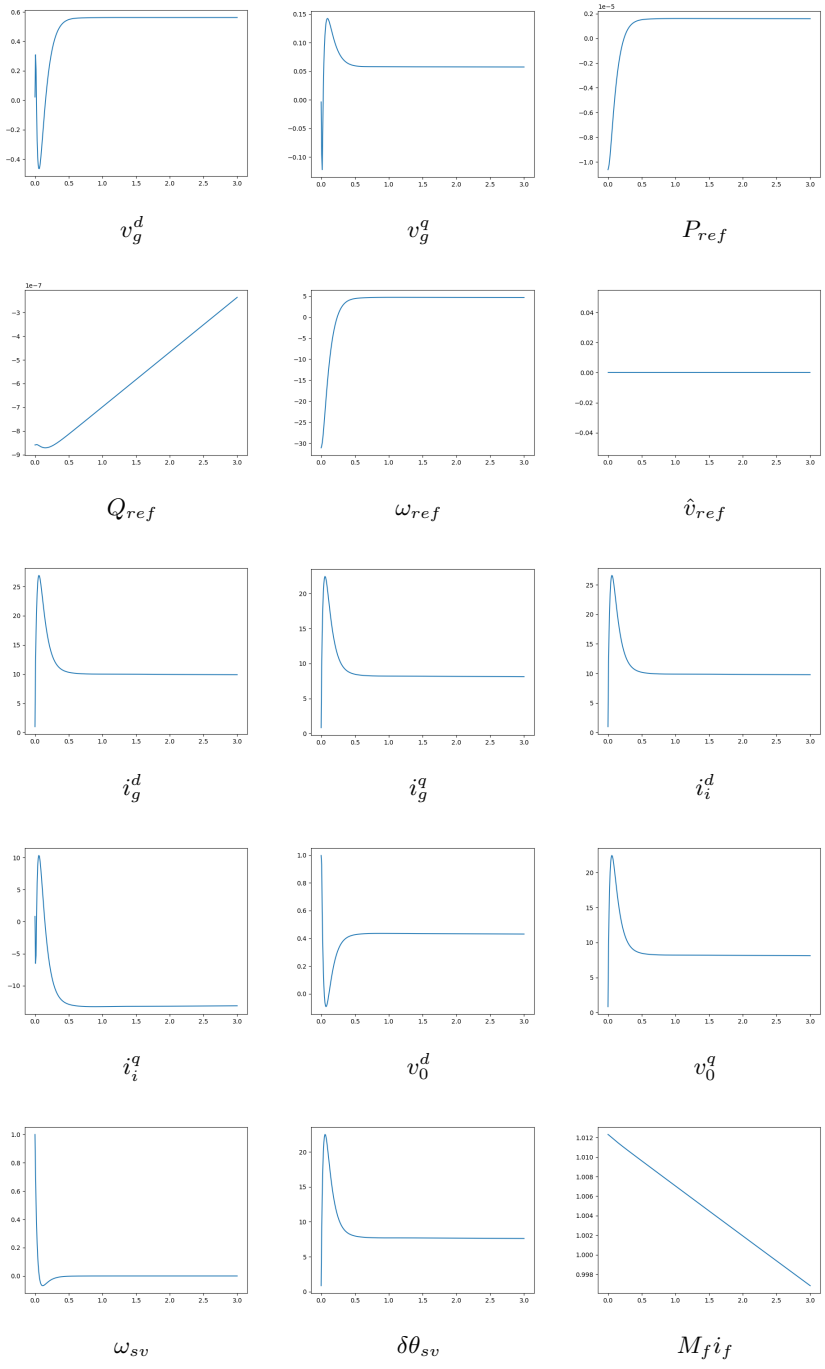

Figure 5: Solution for the states and the optimal controls

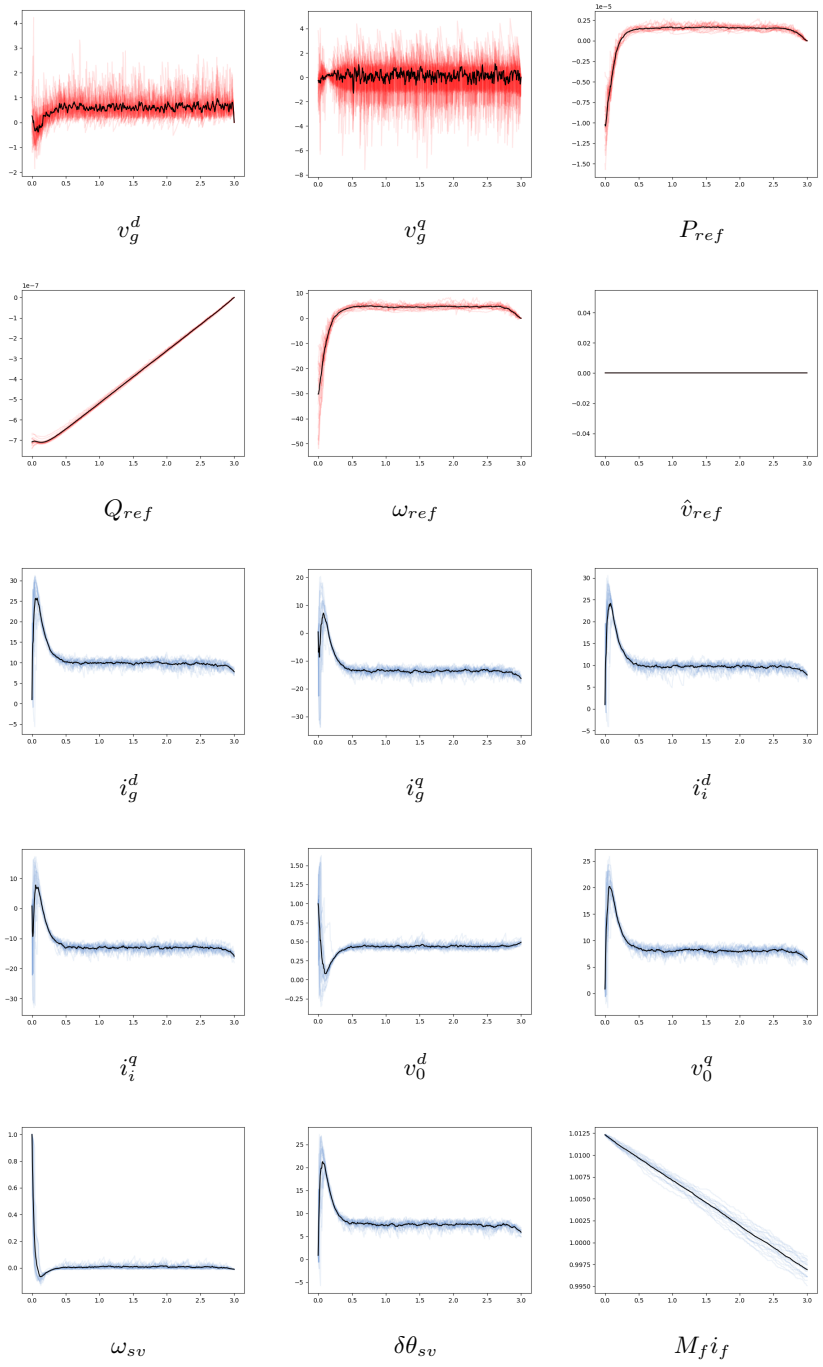

Figure 6: RBM for the states (red) and controls (blue). The black line is the average of the trials in each figure. $K = 1, N = 500$, number of trials 20.

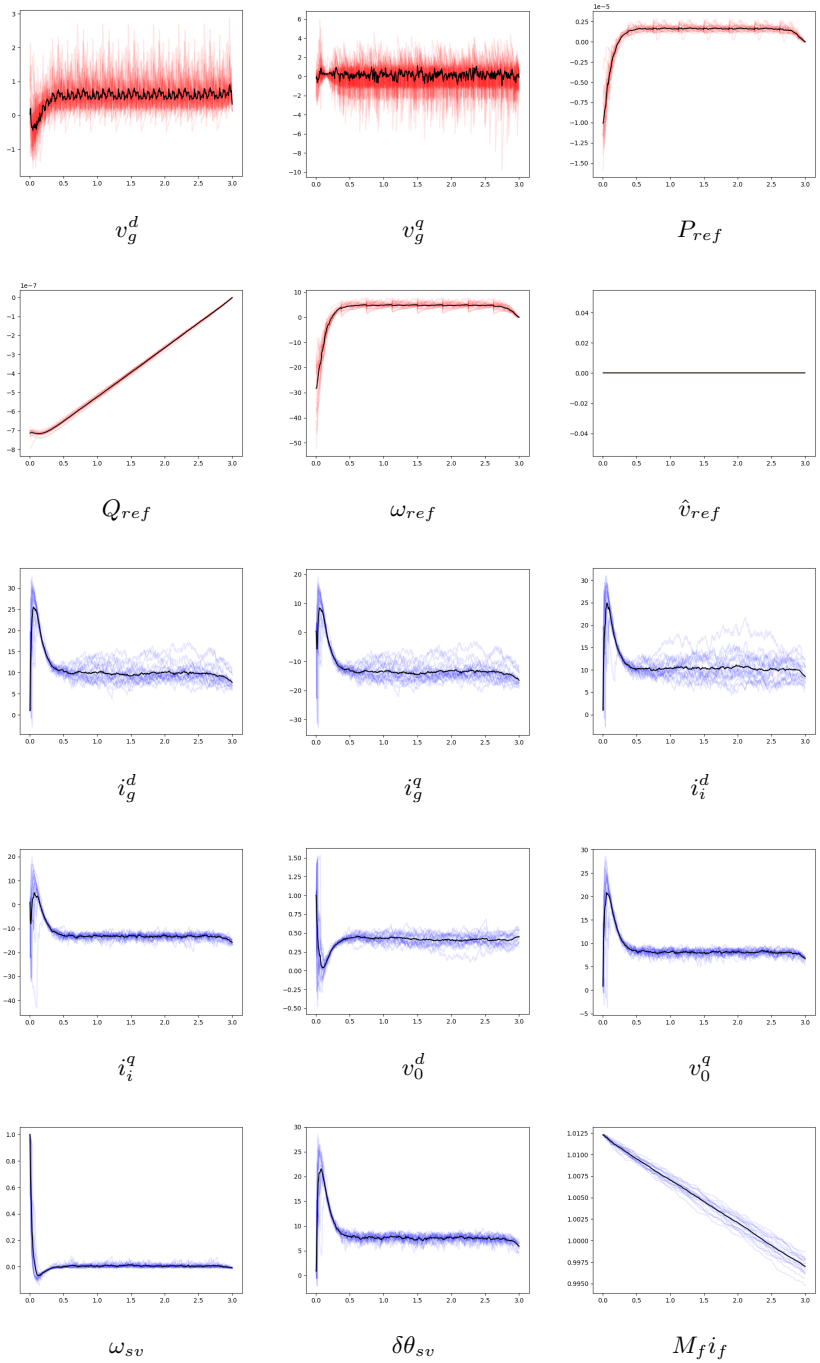

Figure 7: RBM + MPC for the states (red) and controls (blue), with the optimal τ. The black line is the average of the trials in each figure. $K = 1, N = 500$, number of trials 20.

minimizes

$$J_{T,\tau_i}(\boldsymbol{u}(t)) = \int_{\tau_i}^{\tau_i+T} \Big(\boldsymbol{x}(t)^T Q \boldsymbol{x}(t) + \boldsymbol{u}(t)^T R \boldsymbol{u}(t) \Big)\, dt,$$

where $\boldsymbol{x}(t)$ fulfills

$$\dot{\boldsymbol{x}}(t) = A\boldsymbol{x}(t) + B\boldsymbol{u}(t), \quad \boldsymbol{x}(0) = \boldsymbol{x}_0. \tag{8}$$

- We now apply $\boldsymbol{u}^*$ to the true dynamics and obtain, like this, the state $\boldsymbol{x}^*$

$$\dot{\boldsymbol{x}}^*(t) = A\boldsymbol{x}^*(t) + B\boldsymbol{u}^*(t), \quad \boldsymbol{x}^*(0) = \boldsymbol{x}_0, \tag{9}$$

which we set to the MPC trajectory $\boldsymbol{x}^*_{MPC}$ on $t \in [\tau_i, \tau_{i+1}]$:

$$\boldsymbol{x}^*_{MPC}(t) := \boldsymbol{x}^*(t), \quad \text{for } t \in [\tau_i, \tau_{i+1}]. \tag{10}$$

- This procedure is repeated: Starting from the state $\boldsymbol{x}^*_{MPC}(\tau_1)$, we obtain $\boldsymbol{u}^*(t)$ over $[\tau_1, \tau_1+T]$ and consequently $\boldsymbol{x}^*(t)$ over $[\tau_1, \tau_1+T]$, which yields $\boldsymbol{x}^*_{MPC}$ on $[\tau_1, \tau_2]$.

In summary, the MPC algorithm can be described as follows:

1. Initialize the state: $\boldsymbol{x}^*_M(0) = \boldsymbol{x}_0$, $i = 0$.
2. Set the number of steps as $\lceil T/\tau \rceil$.
3. **For** $i \in \{0, 1, 2, ..., \text{steps}\}$:
   a) Compute $\boldsymbol{u}^*(t)$ on $[\tau_i, \tau_i + T]$.
   b) Determine $\boldsymbol{x}^*(t)$ on $[\tau_i, \tau_{i+1}]$ by solving
   $$\dot{\boldsymbol{x}}^*(t) = A\boldsymbol{x}^*(t) + B\boldsymbol{u}^*(t).$$
   c) Set $\boldsymbol{x}^*_{MPC}(t) := \boldsymbol{x}^*(t)$ on $[\tau_i, \tau_{i+1}]$.

7.1 RBM - MPC

In this section, we approximate the optimal trajectories, $\mathbf{x}^(t)$, associated with the optimal control u^* of (2) using the RBM-MPC strategy, which integrates Model Predictive Control (MPC) with the Random Batch Method (RBM). This approach predicts the optimal control for reduced-order dynamics and applies it iteratively. Initially proposed by [6] for nonlinear, multi-agent systems, we adapt this method to our linearized model, as discussed in Section 3.

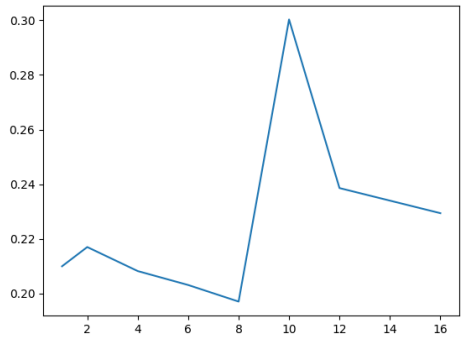

Figure 8: MAE of the RBM-MPC states vs τ

As noted in this Section, the MPC depends of the parameter τ, which can be thought as the number of subintervals into which we divide the time interval $[0, T]$. Therefore, we ought to search the best choice for this parameter, so our simulations are the most accurate. This can be done in two different ways: either we minimize the error with respect to our states or to our controls. We opted to minimize it with respect to the states.

Figure 8 shows the MAE of the RBM-MPC states vs τ. We find that the best choice is $\tau = 8$.

8 Discussion

Dependence of the RBM with respect to N, K and the number of trials

It is evident that as the value of the number of discretization points (N), and the number of trials increase, the RBM achieves more accurate approximations of states and controls. Conversely, when the parameter K is larger, the quality of approximations deteriorates (Figures $9 - 18$).

This observation aligns with the theoretical framework. Let us define $h = K\Delta t$. Obviously, h as an upper bound on the length of the discretized time interval, since $K \geq 1$. The expected error of the RBM is proportional to $\sqrt{h}$ ([7], [11]), therefore the error also is expected to increase with h:

$$E(|x_h(t) - x(t)|) \leq \sqrt{E(|x_h(t) - x(t)|^2)} \leq \sqrt{h\mathrm{Var}(A_h)(||A||t^2 + 2t)(|x_0| + |Bu|_{L^1(0,T;\mathcal{R}^N)})},$$

where in the second inequality, we have used Theorem 1 of [11].

The same goes for the controls (Theorem 4 of [11]):

$$E(|u_h(t) - u(t)|) \leq \sqrt{E(|u_h(t) - u(t)|)^2} \leq \frac{1}{\alpha}\sqrt{C_{[A,B,x_0,Q,R,x_d,T]}h\mathrm{Var}(A_h)},$$

where α is the parameter that defines the $\alpha-$convexity of the functional, which follows from the dynamics being lineal.

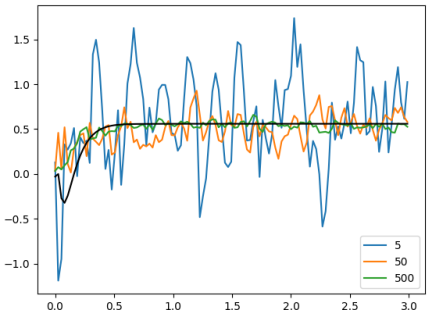

Figure 9: $N = 150, K = 1$

Differences between RBM and MPC

Exploring the relative performance of the RBM and MPC necessitates a detailed examination of error characteristics and their implications for control

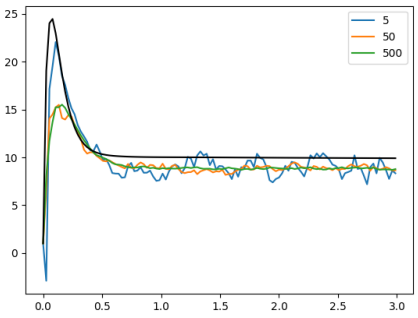

Figure 10: $N = 150, K = 1$

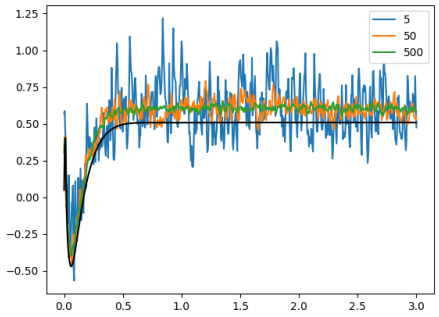

Figure 11: $N = 800, K = 1$

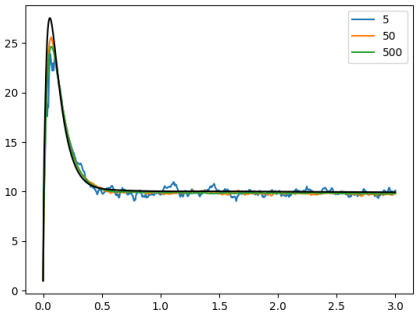

Figure 12: $N = 800, K = 1$

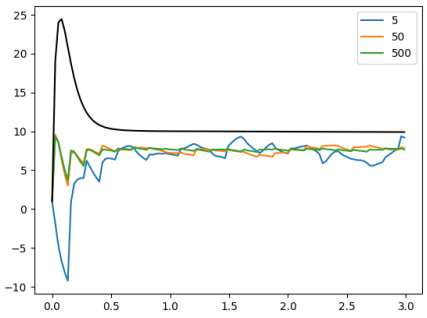

Figure 13: $N = 150, K = 5$

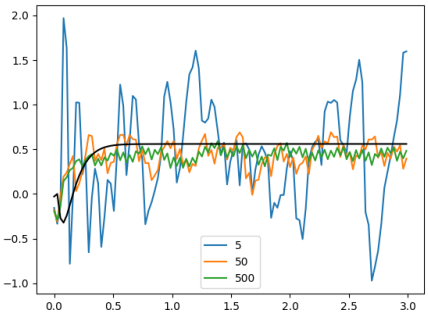

Figure 16: $N = 150, K = 2$

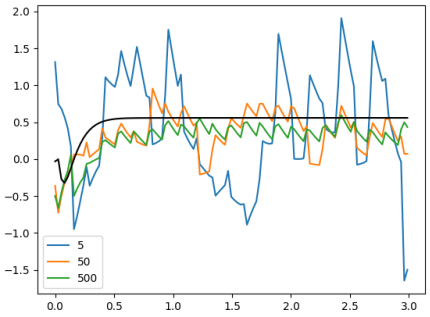

Figure 14: $N = 150, K = 5$

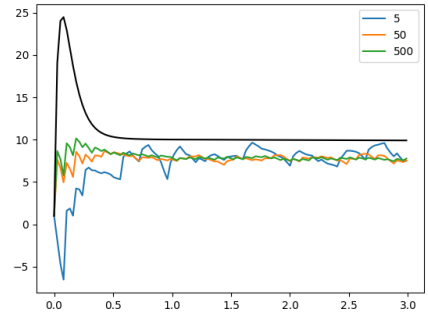

Figure 17: $N = 150, K = 3$

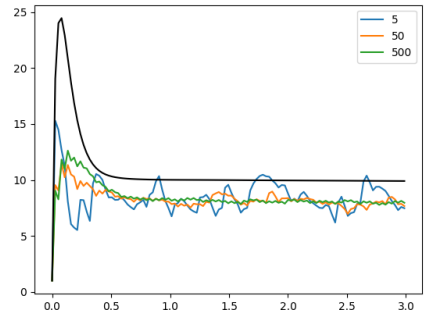

Figure 15: $N = 150, K = 2$

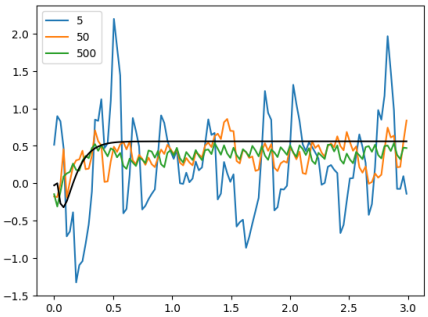

Figure 18: $N = 150, K = 3$

strategies in electrical networks. Drawing upon empirical observations and theoretical insights, we delve into the discrepancies between RBM and MPC, particularly in terms of error dynamics.

The RBM yields reasonable approximations to the original control problem, as corroborated by theoretical studies. Notably, the mean of RBM trials closely approximates the solution, underscoring its efficacy in capturing the system dynamics and the optima controls.

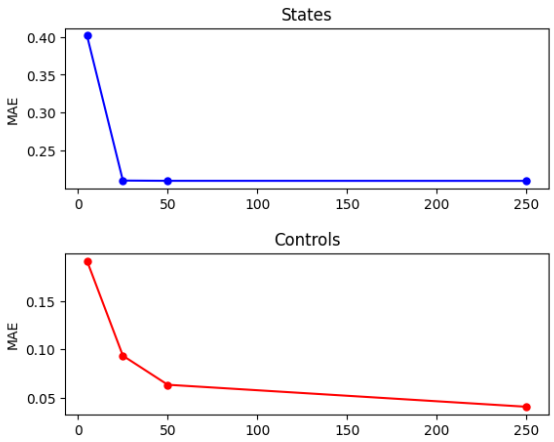

Figure 19: MAE for the states and controls with respect to the number of trials using the RBM.

Similarly, MPC simulations produce comparable, if slightly better results, with the predicted trajectories closely aligning with the desired control objectives. The accuracy of MPC solutions reinforces its utility in devising effective control strategies for electrical networks.

Initial conjectures suggested that MPC might exhibit considerably reduced oscillations compared to RBM due to its piece-wise methodology. However, Figures 6 and 7 show comparable oscillatory behavior between RBM and MPC trials, challenging preconceived notions of MPC's superiority in damping oscillations. To discern the underlying factors contributing to error behavior, simulations were conducted to analyze the discrepancy between the mean of the randomized solutions and the deterministic solution (Figures 19 and 20). Surprisingly, both RBM and MPC exhibited similar error trends, indicating that MPC did not overwhelmingly outperform RBM in this regard. The MAE after 250 trials is, indeed, smaller for the MPC + RBM than for the RBM, but the difference is practically unnoticeable. This parity in error dynamics encourages the need for a nuanced understanding of the underlying mechanisms driving error accumulation in both methodologies.

Despite variations in error characteristics, both RBM and MPC offer viable approaches to control optimization in electrical networks. Understanding the nuances in error behavior is crucial for analyzing the robustness and effectiveness of both methodologies.

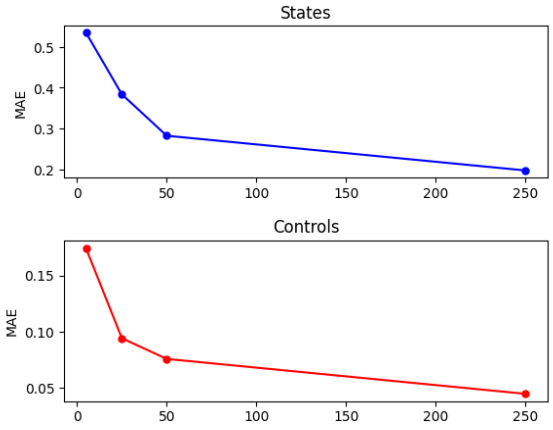

Figure 20: MAE for the states and controls with respect to the number of trials using RBM+MPC, for the optimal τ.

Furthermore, analysis of the MAE unveiled intriguing trends. While both RBM and MPC + RBM initially exhibited satisfactory error levels, a mild increase in error magnitude, particularly for state variables, was observed in the RBM when the number of trials is 250 (Figure 19). This phenomenon hints either at potential numerical challenges or inherent stochasticity behavior, reminiscent of the loss function diminishment in neural networks, possibly attributed to the stochastic nature of the RBM.

Computational time

We examine the computation time of the RBM using three significant examples: the 1D heat equation, the 3D heat equation, and the electrical model detailed in Section 2, which represents the most critical scenario:

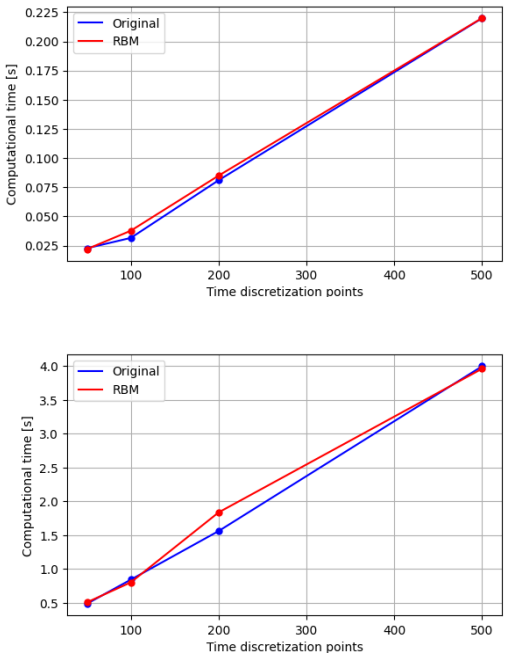

Figure 21: Computational time for the state and control matrices in the 1D heat equation

The left graphs depict the computation time for the state matrix, while the right graphs illustrate the computation time for the control matrix. Notably, the control matrix requires more computation time due to its computation via the gradient algorithm. These three examples were chosen for specific reasons: they demonstrate cases where RBM either reduces computation time (as observed in the 3D heat problem), increases it (as in the electrical model), or has minimal impact on it (as seen in the 1D heat problem) [11].

The reasons behind these varied outcomes lie in two main factors that influence RBM: the structure of the state matrix A and its prior decomposition (splitting). For both the 1D and 3D heat problems, insights from literature suggest that the denser interconnections in the 3D case contribute to its reduced computation time compared to the more straightforward 1D case. Specifically, in the 1D scenario, each spatial node connects with only two neighboring nodes, whereas in 3D, each node connects with six others, resulting in a denser matrix structure A. Moreover, the 3D case benefits significantly from a block-wise decomposition, which substantially reduces computation time (as evident in the second row of figures).

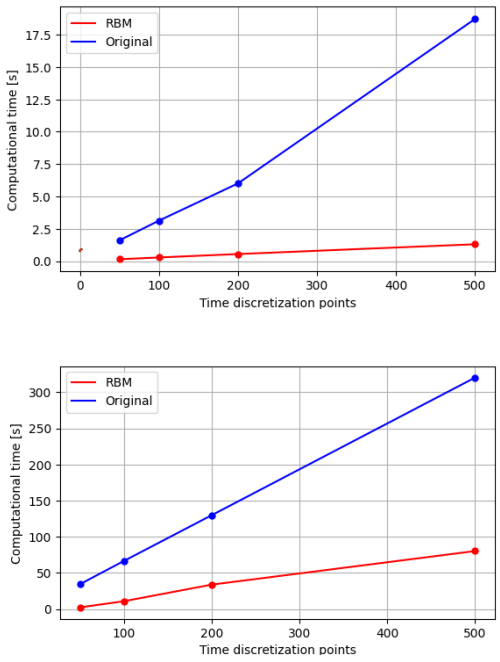

Figure 22: Computational time for the state and control matrices in the 3D heat equation

An intriguing question arises: is it possible to find a way to reduce the 1D case, or does the interconnection structure of the nodes inherently prevent RBM from computationally improving the problem's resolution? Knowing this will help to address the electrical model problem. Notably,

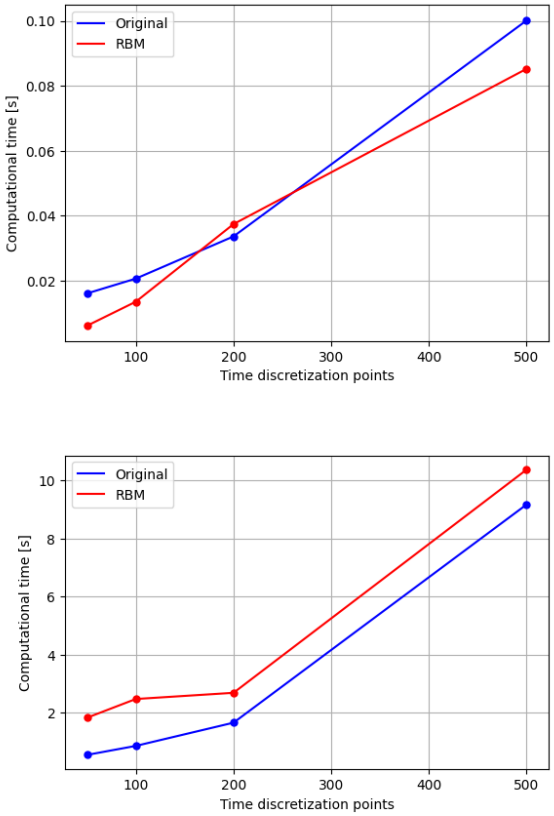

Figure 23: Computational time for the state and control matrices in the electrical model

the increased computational time observed in the electrical model stems from the use of spectral decomposition (Section 5), which results in a denser matrix structure compared to the original. Although the method converges since all submatrices are dissipative, the increased density makes it harder for the computer to perform the calculations. Therefore, this decomposition is not suitable, and another must be found. But what if we encounter a situation similar to the 1D heat case? It could be that the very structure of the electrical model does not allow for sufficiently sparse submatrices to achieve a noticeable reduction. In that case, there are two possibilities: either find a more complex model that permits a matrix where reduction is possible, or move to a three-dimensional electric PDE model, analogous to the 3D heat problem, where better results are obtained.

In the next Subsection, we explore other block-wise decompositions to see if it is possible to solve these problems, while mantaining a small error.

Other decompositions

The deterministic approach

We can search for other block decompositions to approximate the original system. The initial block decomposition (Section 5) did not adequately approximate the original problem. Presumably, this occurred because most of the submatrices were empty, resulting in solutions that did not effectively approximate the original problem. To address this, we added the zero submatrices to the other submatrices, creating a new, shorter array of submatrices where none of them are zero. Additionally, we combined them in such a way that none of their eigenvalues are positive.

The resulting approximations, while improved compared to the previous block decomposition, still exhibit limitations. The main challenge lies in the persistence of stiffness within the reduced matrix systems, thereby complicating convergence for gradient-based algorithms unless the number

of discretization points is substantially increased. Moreover, due to this block decomposition approach, subsystems often exhibit increased stiffness compared to the original solution, necessitating a higher number of discretization points. For instance, with the initial decomposition, a minimum of $K = 300$ points are needed for the gradient to descend and converge. Figure 24 shows the first state and the first control as an example.

The RBM is somewhat similar to the original solution, but not quite accurate. Things get worse in the last state, where the RBM is totally different (Figure 24):

Therefore, while this decomposition approach presents some improvements, it remains the best block decomposition identified thus far. However, this does not imply that an optimal block decomposition does not exist; rather, it underscores the need to identify one where all subsystems associated with the submatrices exhibit similar stiffness characteristics as the original. This consideration is crucial, especially for future computational time analyses, as a valid decomposition that complicates gradient descent algorithms may increase computation time. This aspect remains underexplored in the literature, highlighting the significance of addressing additional challenges posed by stiff systems.

The non-deterministic approach: random block-wise decomposition

In this method, the state matrix A is decomposed into 3x3 blocks using a randomized approach rather than a deterministic one, based on adjacent blocks. The procedure follows these steps:

1. First, I permute the rows and columns of A, simply permuting the set $\{1, 2, ..., 9\}$ twice:

$$A' = PAQ$$

where P and Q are permutation matrices derived from the set $\{1, 2, ..., 9\}$.

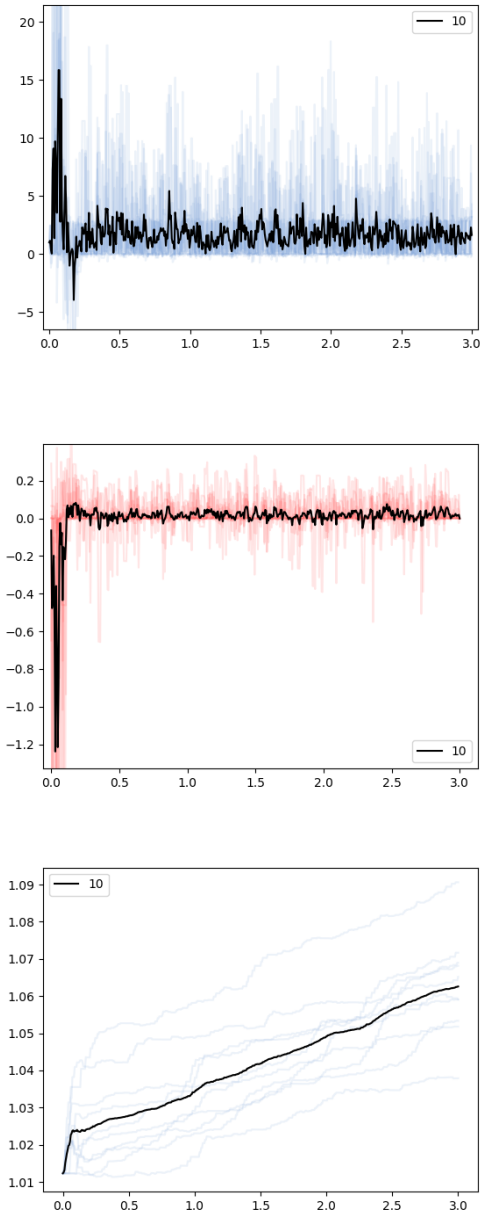

Figure 24: RBM for the first and last state and the first control

2. Then, I proceed to decompose by adjacent blocks, but this is done based on the new permuted matrix:

$$A' = \begin{bmatrix} A'_{11} & A'_{12} & A'_{13} \\ A'_{21} & A'_{22} & A'_{23} \\ A'_{31} & A'_{32} & A'_{33} \end{bmatrix}$$

where each A'_{ij} is a 3x3 block.

3. Reintroduce each 3x3 block back into a new 9x9 matrix B, preserving its position from the permuted matrix:

$$B_{ij} = A'_{ij}$$

Thus, B is formed by reassembling the 3x3 blocks from A' into the positions they correspond to in A.

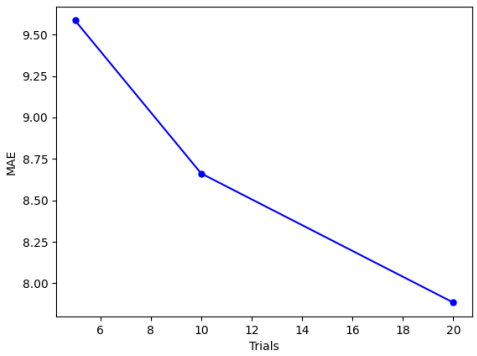

Figure 25: MAE for the states using the non-deterministic approach of the prior decomposition of A

This approach, however, did not lead to significant improvements, since the MAE for the states is approximately 8 for 20 trials (Figure 25).

Making the system more complex

Here we assess the of RBM and MPC methodologies when we increase the size of the system. In order to do that, we just plug two more converters and two more regulators (see Apendix B), making the state matrix A 19x19:

$$A = \begin{bmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,18} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,18} \\ \vdots & \vdots & \ddots & \vdots \\ A_{18,0} & A_{18,1} & \cdots & A_{18,18} \end{bmatrix}$$

We tested the two mentioned decompositions: the *spectral* and the *block-wise* decomposition.

Spectral decomposition

We see that the RBM did not decrease the computational time, even when the size of the system has increased (Figure). The MAE, however, is much bigger now for the states (Figure 26).

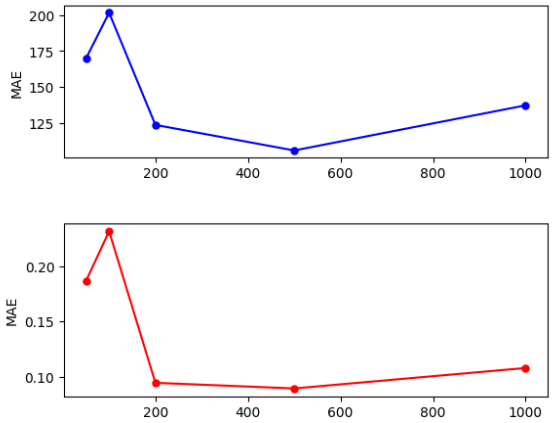

Figure 26: MAE for the states and controls with respect to the number of trials using the RBM.

Block-wise decomposition

Next, we proceed to the decomposition of a 19x19 matrix into various non-overlapping blocks. Since A is no more square, we have to modify the methodology of splitting A. The blocks used in this decomposition are of sizes 5x5, 5x4, 4x5, and 4x4.

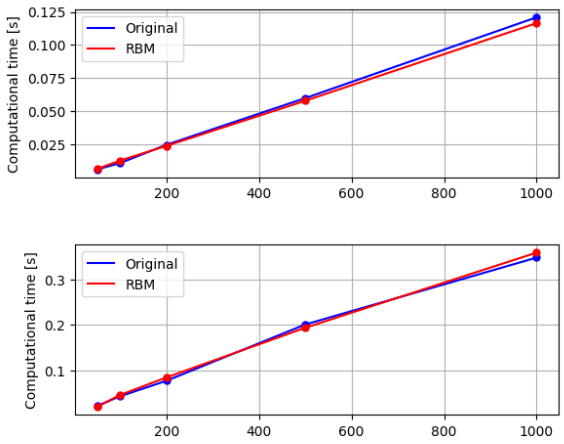

Figure 27: Computational time for the state and control matrices

5x5 Blocks

The 5x5 blocks are placed in the top-left of the matrix:

$$\text{Block}_{5x5} = \begin{bmatrix} A_{0,0} & A_{0,1} & A_{0,2} & A_{0,3} & A_{0,4} \\ A_{1,0} & A_{1,1} & A_{1,2} & A_{1,3} & A_{1,4} \\ A_{2,0} & A_{2,1} & A_{2,2} & A_{2,3} & A_{2,4} \\ A_{3,0} & A_{3,1} & A_{3,2} & A_{3,3} & A_{3,4} \\ A_{4,0} & A_{4,1} & A_{4,2} & A_{4,3} & A_{4,4} \end{bmatrix}$$

$$\text{Block}_{5x4} = \begin{bmatrix} A_{0,15} & A_{0,16} & A_{0,17} & A_{0,18} \\ A_{1,15} & A_{1,16} & A_{1,17} & A_{1,18} \\ A_{2,15} & A_{2,16} & A_{2,17} & A_{2,18} \\ A_{3,15} & A_{3,16} & A_{3,17} & A_{3,18} \\ A_{4,15} & A_{4,16} & A_{4,17} & A_{4,18} \end{bmatrix}$$

4x5 Blocks

The 4x5 blocks fill the bottom-left corner:

$$\text{Block}_{4x5} = \begin{bmatrix} A_{15,0} & A_{15,1} & A_{15,2} & A_{15,3} & A_{15,4} \\ A_{16,0} & A_{16,1} & A_{16,2} & A_{16,3} & A_{16,4} \\ A_{17,0} & A_{17,1} & A_{17,2} & A_{17,3} & A_{17,4} \\ A_{18,0} & A_{18,1} & A_{18,2} & A_{18,3} & A_{18,4} \end{bmatrix}$$

4x4 Blocks

Finally, the 4x4 blocks occupy the bottom-right corner:

$$\text{Block}_{4x4} = \begin{bmatrix} A_{15,15} & A_{15,16} & A_{15,17} & A_{15,18} \\ A_{16,15} & A_{16,16} & A_{16,17} & A_{16,18} \\ A_{17,15} & A_{17,16} & A_{17,17} & A_{17,18} \\ A_{18,15} & A_{18,16} & A_{18,17} & A_{18,18} \end{bmatrix}$$

The results for that decomposition were analogous that of the simplest model: the RBM *did not* converge, since the functional explodes for any number of discretization points.

9 Open Problems and Perspectives

While the RBM and MPC show promise in addressing uncertainties in Converter-Dominated Power Systems, several open problems and avenues for future research exist.

9.1 Computational Cost Reduction with RBM

It remains to be seen whether the RBM effectively reduces the computational cost of electrical network models. While the RBM shows promise in providing reasonable approximations to the original control problem (2)-(3), its impact on computational efficiency requires further investigation. Particularly, assessing RBM's performance in more complex systems is essential to ascertain its potential for decreasing computational burden, especially concerning the MPC, which exhibits considerable execution time.

9.2 Treatment of Stiff Problems

Dealing with stiff problems poses a significant challenge for both RBM and MPC algorithms. The lack of clarity regarding the prior decomposition of matrices and the selection of adaptive step methods exacerbates computational complexity and runtime. Overcoming these hurdles requires devising robust algorithms capable of efficiently handling stiff systems

while maintaining computational tractability. Additionally, exploring alternative strategies to improve convergence rates and mitigate computational time in stiff problem formulations is imperative for enhancing the efficacy of RBM and MPC in addressing linear and non-linear electrical network dynamics.

9.3 Discrepancy Between Theory and Practice

The stark contrast between control theory and practical implementation presents notable challenges, primarily attributed to numerical difficulties. While theoretical frameworks offer insights into optimal control strategies, translating these concepts into practical applications is often conditioned by numerical complexities and computational limitations. Bridging the gap between theory and practice necessitates the development of robust algorithms and methodologies capable of addressing real-world constraints while maintaining mathematical rigor.

Conclusions

The RBM and MPC demonstrate significant promise in handling the control of Converter-Dominated Power Systems. This study pioneers the assessment of Data-based Reduced Order Models, specifically RBM and MPC, in managing stiff systems. Our findings underscore their potential but also highlight challenges that warrant further exploration. With the simulations we have carried out, we can state that:

- The inadequacy of the approximations to accurately capture the solution suggests a fundamental limitation: the non-sparsity of the state matrix A (See the Appendix) seems to hinder its effective division into blocks. This observation challenges the applicability of block-wise strategies in this context, since the approximations to the deterministic solutions with this decomposition are not accurate (Figures 2 and 3).

- Managing the time splitting method poses challenges, particularly in the context of non-sparse matrices, where determining the most suitable technique for the prior decomposition of the state matrix A is not straightforward. Although *spectral* decomposition has shown promise in yielding favorable results (Figures 5−20), the quest for an alternative, potentially more efficient method to partition A remains unresolved. The selection of the optimal method for decomposing A is a pivotal factor in improving the overall efficacy of the time splitting approach.

- Combining the Model Predictive Control (MPC) with the Random Batch Method (RBM) is a promising technique to address optimal control problems constrained by state trajectories. This method involves breaking down the time into smaller sub-intervals, enabling the efficient resolution of problems, particularly when dealing with a sizable time horizon T. We observed that our initial belief that the MPC would lead to smoother control trajectories compared to the RBM was proven wrong, since Figures 19 and 20 showed similar error control trends in both RBM and MPC. Moreover, when we compared the mean of randomized solutions to the deterministic solution, we found that the improvement of the RBM-MPC over the RBM is very marginal, which does not compensate for the increased computation time. These findings emphasize the importance of understanding the factors influencing error accumulation in both methods.

- The relationship between the RBM, the number of discretization points N, and the switching parameter K is evident: an increase in these parameters leads to enhanced accuracy in approximations for states and controls. However, a larger value of K is associated with a deterioration in the quality of these approximations, in line with the analytical results.

References

[1] Umberto Biccari, Noboru Sakamoto, Eneko Unamuno, Danel Madariaga, Enrique Zuazua, and

Jon Andoni Barrena. Model Reduction of Converter-Dominated Power Systems by Singular Perturbation Theory. *arXiv preprint arXiv:1910.09222*, 2019.

[2] Lars Grüne, Jürgen Pannek, Lars Grüne, and Jürgen Pannek. *Nonlinear model predictive control*. Springer, 2017.

[3] Yunjie Gu, Nathaniel Bottrell, and Timothy C Green. Reduced-order models for representing converters in power system studies. *IEEE Transactions on Power Electronics*, 33(4):3644–3654, 2017.

[4] Shi Jin, Lei Li, and Jian-Guo Liu. Random batch methods (RBM) for interacting particle systems. *Journal of Computational Physics*, 400:108877, 2020.

[5] Shi Jin, Lei Li, and Jian-Guo Liu. Convergence of the random batch method for interacting particles with disparate species and weights. *SIAM Journal on Numerical Analysis*, 59(2):746–768, 2021.

[6] Dongnam Ko and Enrique Zuazua. Model predictive control with random batch methods for a guiding problem. *Mathematical Models and Methods in Applied Sciences*, 31(08):1569–1592, 2021.

[7] Lei Li, Zhenli Xu, and Yue Zhao. A random-batch Monte Carlo method for many-body systems with singular kernels. *SIAM Journal on Scientific Computing*, 42(3):A1486–A1509, 2020.

[8] Benjamin Peherstorfer, Serkan Gugercin, and Karen Willcox. Data-driven reduced model construction with time-domain loewner models. *SIAM Journal on Scientific Computing*, 39(5):A2152–A2178, 2017.

[9] Benjamin Peherstorfer and Karen Willcox. Dynamic data-driven reduced-order models. *Computer Methods in Applied Mechanics and Engineering*, 291:21–41, 2015.

[10] Daniel Veldman. Random Batch Methods for Linear-Quadratic Optimal Control Problems. http://dcn.nat.fau.eu/random-batch-methods-for-linear-quadratic-optimal-con 2021.

[11] Daniel WM Veldman and Enrique Zuazua. A framework for randomized time-splitting in linear-quadratic optimal control. *Numerische Mathematik*, 151(2):495–549, 2022.

[12] Qing-Chang Zhong. Power-electronics-enabled autonomous power systems: Architecture and technical routes. *IEEE Transactions on Industrial Electronics*, 64(7):5907–5918, 2017.

[13] Qing-Chang Zhong, Phi-Long Nguyen, Zhenyu Ma, and Wanxing Sheng. Self-synchronized synchronverters: Inverters without a dedicated synchronization unit. *IEEE Transactions on power electronics*, 29(2):617–630, 2013.

[14] Qing-Chang Zhong and George Weiss. Synchronverters: Inverters that mimic synchronous generators. *IEEE transactions on industrial electronics*, 58(4):1259–1267, 2010.

State and control Matrices

$$A = \begin{pmatrix} -15.45 & 57.17 & 0 & 0 & 2128.99 & 0 & 0 & 0 & 0 \\ -1726.44 & -15.45 & 0 & 0 & 0 & 2078.63 & 0 & 0 & 0 \\ 0 & 0 & -11.82 & 57.17 & -5322.49 & 0 & 5272.42 & -266.42 & 5208.30 \\ 0 & 0 & -1726.44 & -11.82 & 0 & -5196.58 & 6100.30 & 6953.81 & 6026.11 \\ -3881.86 & 0 & 3881.86 & 0 & 0 & 55.82 & 0 & 0 & 0 \\ 0 & -3975.91 & 0 & 3975.91 & -1768.27 & 0 & 0 & 0 & 0 \\ 0 & 0 & -0.06 & -0.00 & -0.06 & -0.00 & -41.77 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1308.99 & 0 & 0 \\ 0 & 0 & -0 & 0 & 0 & -0 & 0 & -0 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} -3.78 & 0 & 0 & 0 & 0 & 0 \\ 0 & -20.79 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.13 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

General formulation for the Converter-Dominated Power System Model

$$\begin{cases} \frac{di_g}{dt} & = \frac{1}{L_g}(v_0 - R_g i_g - v_g) - j\omega_g i_g \\ \frac{di_i^k}{dt} & = \frac{1}{L_f^k}(\mathbf{m^k} V_{dc}^k - R_f i_i^k - v_0) - j\omega_g i_i^k \\ \frac{dv_0}{dt} & = \frac{1}{C_T}(i_T - i_g) - j\omega_g v_0 \\ \frac{d(M_f i_f^k)}{dt} & = \frac{1}{K^k}\left[Q_{ref} - Q^k + K_q(\hat{v}_{ref}^k - \hat{v}_0^c)\right] \\ \frac{d\omega_{sv}^k}{dt} & = \frac{1}{T_a^k}\left[\frac{P_m^k}{\omega_{sv}^k} - \frac{P_e^k}{\omega_{sv}^k} - K_D(\omega_{sv}^k - \omega_{ref}^k)\right] \\ \frac{d\delta\theta_{sv}^k}{dt} & = \omega_{sv}^k - \omega_g, \end{cases} \tag{13}$$

where $C_T = \sum_k C_f^k$ and $i_T = \sum_k i_f^k$. Note that $k \in \{1, 2, ..., n\}$ represents the index of each converter. This notation is used to differentiate between multiple converters in the system, allowing for a clear representation of their individual variables.