

## Article

# Switched Auto-Regressive Neural Control (S-ANC) for Energy Management of Hybrid Microgrids

Muhammed Cavus <sup>1,2,\*</sup> , Yusuf Furkan Ugurluoglu <sup>1,3</sup>, Huseyin Ayan <sup>1,4</sup>, Adib Allahham <sup>5</sup>, Kabita Adhikari <sup>1</sup> and Damian Giaouris <sup>1</sup>

<sup>1</sup> School of Engineering, Newcastle University, Newcastle upon Tyne NE1 7RU, UK; y.f.ugurluoglu2@newcastle.ac.uk (Y.F.U.); h.ayan2@newcastle.ac.uk (H.A.); kabita.adhikari@newcastle.ac.uk (K.A.); damian.giaouris@newcastle.ac.uk (D.G.)

<sup>2</sup> The School of Engineering, Iskenderun Technical University, Iskenderun 31200, Turkey

<sup>3</sup> Department of Mechanical Engineering, Necmettin Erbakan University, Konya 42080, Turkey

<sup>4</sup> The School of Engineering, Istanbul University-Cerrahpasa, Istanbul 34320, Turkey

<sup>5</sup> Faculty of Engineering and Environment, Northumbria University, Newcastle NE1 8ST, UK; adib.allahham@northumbria.ac.uk

\* Correspondence: m.cavus2@newcastle.ac.uk

**Abstract:** Switched model predictive control (S-MPC) and recurrent neural networks with long short-term memory (RNN-LSTM) are powerful control methods that have been extensively studied for the energy management of microgrids (MGs). These methods ease constraint satisfaction, computational demands, adaptability, and comprehensibility, but typically one method is chosen over the other. The S-MPC method dynamically selects optimal models and control strategies based on the system's operating mode and performance objectives. On the other hand, integration of auto-regressive (AR) control with these powerful control methods improves the prediction accuracy and the adaptability of the system conditions. This paper compares the two control approaches and proposes a novel algorithm called switched auto-regressive neural control (S-ANC) that combines their respective strengths. Using a control formulation equivalent to S-MPC and the same controller model for learning, the results indicate that pure RNN-LSTM cannot provide constraint satisfaction. The novel S-ANC algorithm can satisfy constraints and deliver comparable performance to MPC, while enabling continuous learning. The results indicate that S-MPC optimization increases power flows within the MG, resulting in efficient utilization of energy resources. By merging the AR and LSTM, the model's computational time decreased by nearly 47.2%. In addition, this study evaluated our predictive model's accuracy: (i) the R-squared error was 0.951, indicating a strong predictive ability, and (ii) mean absolute error (MAE) and mean square error (MSE) values of 0.571 indicate accurate predictions, with minimal deviations from the actual values.

**Keywords:** auto-regressive; control and optimization; energy management; recurrent neural network; long short-term memory; microgrid; switched model predictive control



**Citation:** Cavus, M.; Ugurluoglu, Y.F.; Ayan, H.; Allahham, A.; Adhikari, K.; Giaouris, D. Switched Auto-Regressive Neural Control (S-ANC) for Energy Management of Hybrid Microgrids. *Appl. Sci.* **2023**, *13*, 11744. <https://doi.org/10.3390/app132111744>

Academic Editor: Frede Blaabjerg

Received: 12 September 2023

Revised: 19 October 2023

Accepted: 23 October 2023

Published: 26 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Model predictive control (MPC) is a control approach that is widely utilized in many industries, including chemical, electrical, and mechanical engineering. It is well-suited to microgrids (MGs) because it deals with restrictions and optimizes performance over time [1–5]. MPC entails formulating and solving an optimization problem at each time step, to determine the optimal control inputs for the next step. A MPC was described in [6–9] for effective MG optimization, and mixed integer linear programming (MILP) was employed to solve the problem posed. An MPC-inspired energy management (EM) system employing a neuro-fuzzy method, which accounted for renewable energy sources (RES's) intermittent nature in grid-connected MG with loads and photovoltaic (PV) sources, was reported in [10]. Ref. [11] presented scenario-based stochastic programming with a rolling horizon

strategy for minimizing the operating expenses of MGs when the wind speed is unknown. Rolling horizon or MPC techniques are reactive-based methodologies that modify or update data using deterministic approaches. A scenario-based MPC was developed in [12] to reduce operating expenses and overall emissions. To achieve inexpensive and flexible operation, Ref. [13] provided an MPC-based optimum management for renewable energy MGs with hybrid energy storage systems (ESSs), such as hydrogen, batteries, and capacitors. A hierarchical MPC-based technique for islanded AC MGs addressed power-quality and unbalanced power-sharing difficulties [14]. Despite this, the traditional MPC cannot control a MG in various operational modes.

In contrast, switched model predictive control (S-MPC) is a variant of MPC that employs multiple models, each representing a unique mode of operation or scenario of the system. S-MPC selects the optimal model and associated control strategy based on the current system state and the desired performance goals. This makes it possible for S-MPC to handle systems with mode-dependent dynamics. MPC is distinguished from S-MPC in using a single model to predict the system's future behavior [15]. S-MPC employs multiple models and switches between them, based on the system's current state. S-MPC can provide better performance and robustness than MPC, especially for complex systems with multiple modes or operating conditions [16,17]. Another novel study presented a hybrid MG model that incorporated two switched receding horizon control laws. This strategy reduces the overall energy expenses and maximizes the efficient utilization of RESs for expansive business establishments, while accommodating fluctuations in grid connectivity [18]. Moreover, Ref. [19] outlined the process of designing and applying a S-MPC to wind turbine systems, intending to manage the intricate nature and nonlinearity inherent in wind turbine systems. The system employs qpOASES as an integrated solver for optimum online control. It incorporates a cyber-physical real-time emulator for utility-scale wind turbines, with variable-speed and variable-pitch capabilities. The study showcased the viability and efficacy of S-MPC in attaining control objectives for wind turbine systems in real-time, utilizing brief control periods. In addition, the study in [20] presented a novel technique for enhancing wind turbine control by introducing a S-MPC framework. The proposed approach aimed to solve the limitations of the conventional continuous control-based MPC algorithm. The results of the comparative analysis indicated that the proposed algorithm exhibited superior performance compared to the existing MPC in various aspects, including computational efficiency, load mitigation, and dynamic response. Ref. [21] presented a novel S-MPC method specifically tailored to discrete-time nonlinear systems. The simulation outcomes emphasized its superiority over a conventional MPC technique regarding computational efficacy and control effectiveness. Another study presented a novel S-MPC methodology for power converters. During transient periods, the system utilized horizon-one nonlinear finite control set MPC to steer the system towards the intended reference [22].

On the other hand, S-MPC performance is highly vulnerable to model mismatch. In other words, it must select a suitable system model. Furthermore, the increased complexity of S-MPC impacts the stability and maintainability of MG control [23,24]. These challenges lead to accuracy issues with S-MPC methods. In addition, the computational time of S-MPC is much longer, because of the prediction horizon and various steps. Many authors have studied machine learning (ML) techniques to increase the accuracy of MG systems.

When improving the scheduling effectiveness in networked microgrids (NMGs), the main goal is minimizing the effects of electricity outages. This paper presents a framework consisting of three stages, to evaluate power transactions, manage renewable energy and market price risks, and tackle uncertainties. This framework is formulated as a mixed-integer linear programming problem [25–28]. On the other hand, ref. [29] introduced a novel approach utilizing the Internet of things (IoT) to dynamically optimize and regulate power loads in citizen energy communities. This technique is compared to the conventional direct load control (DLC) method. This technique aims to enhance power use efficiency using programmable appliances and dynamic demand response. In order to model the

behavior of RESs, such as wind and solar, auto-regressive moving-average (ARMA)-based scenario generation was implemented. Large industries could receive direct assistance from storage and demand-side management systems, to reduce energy costs [30]. The other work employed an ARMA model to forecast solar PV, wind power generation, and electricity demand. Second, an optimal generation scheduling procedure is intended to reduce system operating expenses. The simulation results indicated that the optimal generation scheduling could minimize operating expenses under the worst-case scenario [31]. In [32], combining two models, the ARMA and nonlinear auto regressive with exogenous input (NARX), a novel method was presented for predicting solar radiation. The decision was made to utilize the benefits of both models, to produce more accurate prediction results. The simulation results validated this hybrid model's ability to predict weekly solar radiation averages. Although the previous solar radiation forecasting techniques, particularly ARMA models, are effective for particular uses, they are unsuitable for other uses requiring a high forecasting precision. Several researchers have proposed hybrid models to improve the precision of solar radiation forecasting. Moreover, there is still not a proper plant model or prediction horizon, so the computational time of the model is still to high [32].

There have been numerous studies on using ML methods in place of AR models. For instance, ref. [33] thoroughly investigated the prediction performance of several recurrent neural network (RNNs) designs, such as a long short-term memory (LSTM), gated recurrent unit (GRU), and bidirectional LSTM. Using local weather forecasts and historical weather data, ref. [34] proposed a LSTM-based next-day forecasting model of hourly global horizontal irradiance (GHI). Refs. [35,36] suggested LSTM-based models using only the next day's weather forecast as input. The studies by [37,38] used similar LSTM-based techniques. Refs. [39,40] validated the performance of hybrid deep learning models built on convolutional neural networks (CNNs) and LSTM for day-ahead GHI forecasting. In addition to RNN-based approaches, there have been studies evaluating the performance of other statistical and ML models for solar irradiance forecasting, such as coupled AR and a dynamic system by [41], a Markov switch model [42], and a support vector machine (SVM) by [43]. Ref. [44] reported an LSTM-based model for hour-ahead solar irradiance forecasting. The inputs, which included historical GHI and meteorological data from the preceding 24 h, were utilized to forecast the GHI for the next hour [44]. The results revealed that the LSTM-based model outperformed the other models, such as auto-regressive integrated moving average (ARIMA) and CNN [44]. Ref. [45] investigated the performance of LSTM and GRU. Refs. [46,47] published hybrid CNN-LSTM models for hour-ahead GHI forecasting. Their study showed that incorporating external weather information considerably increased the prediction accuracy. Unlike day-ahead irradiance forecasting methods, hour-ahead forecasting algorithms create projections for the following hour only using historical data.

On the other hand, RNNs are a form of ML technology widely employed for time series prediction and the modeling of dynamic systems [48,49]. RNNs are artificial neural networks (ANN) that are particularly useful for modeling time-series data and may be used to anticipate future MG behavior [50,51]. RNNs may learn and adapt to system dynamics by learning the temporal dependencies in data. RNNs have been used to solve various MG control challenges, including load forecasting, renewable energy integration, and demand response management [52–54]. RNNs have been applied to various systems, including power systems [55,56], with promising prediction accuracy and flexibility results.

In summary, both control families have benefits and drawbacks, and their complementarity is evident. On the one hand, S-MPC struggles with system complexity and long-term prediction horizons, whereas the combination of AR and LSTM (AR-LSTM) can deal with complex systems and infinite prediction horizons. AR-LSTM, conversely, has difficulty in satisfying constraints and lacks interpretability, whereas S-MPC can provide safety guarantees and understandability.

Although there is a clear potential for a synergy between the two families of methods, there have been few attempts to combine their relative advantages. This research deficiency

is not limited to applying EM for MG. The control and ML communities evolve independently, adopting radically different notations to formulate the same problem. In spite of the parallel developments, several authors [57–59] have suggested that a collaboration between the two groups could result in potential advantages. Combining these methodologies represents a powerful method for integrating robust control theory methods with ML approaches, to exploit additional information from real-time data [60,61].

As shown in Table 1, each control method has its strengths and limitations. MPC and S-MPC offer robust optimality and constraint handling but may have computational challenges. AR and RNN-LSTM are efficient in computation but may not manage complex constraints effectively. S-ANC combines AR models with neural control, balancing optimality and computational efficiency. The choice of control method depends on the specific application and trade-offs between these criteria.

**Table 1.** Comparison of Control Methods.

Control Method	Optimality	Computational Time [s]	Multiple Models	Adaptability	Constraints
MPC [62,63]	✓	$>10^4$ [High]	×	Good	✓
S-MPC [15,16,20,64]	✓	$\approx 10^4$ [Moderate]	✓	Outstanding	✓
DLC [29]	✓	$\approx 10^4$ [Moderate]	×	Good	✓
AR [32]	×	$\approx 10^4$ [Moderate]	×	Poor	×
CNN [65,66]	×	$<10^4$ [Low]	×	Poor	×
RNN-LSTM [67,68]	✓	$<10^4$ [Low]	×	Poor	×
S-ANC	✓	$< 10^4$ [Low]	✓	Outstanding	✓

#### Contributions and Research Questions

This paper was motivated by how AR-LSTM and S-MPC could collaborate in applying the EM of MGs. While there is a consensus that combining the two algorithms may yield benefits, little has been done to develop methods that involve the two algorithms working together. In addition, works have investigated how these controllers can collaborate with the algorithms working with different control designs and modes. No previous research has compared and combined S-MPC and AR-LSTM for the same optimal control problem formulation in EM for MGs.

The second objective of this paper was to propose a novel method known as switched auto-regressive neural control (S-ANC), which merges S-MPC and AR-LSTM synergistically. The development and formulation of this new S-ANC algorithm were motivated by the conceptual and practical comparison of S-MPC and AR-LSTM. In contrast to comparable approaches, our method combines the S-MPC objective function and constraints with the AR-LSTM optimization and prediction function. This practice ensures interoperability between the two methods and enables the truncation of the S-MPC optimization problem, which can become highly complex, even for relatively simple MG structures. Finally, a flexible hybrid MG case is used to describe and evaluate this new algorithm.

Consequently, the primary contribution of this paper is the introduction of S-ANC, a control algorithm that combines techniques from the communities of control theory and ML. This algorithm is evaluated, and a new standard framework is generated for EM of hybrid MGs. In addition, the proposed S-ANC algorithm applies to various applications and domains, such as complex industrial processes and energy markets. This study also combines control theory and ML by comparing and disentangling the key distinctions between S-MPC and AR-LSTM.

## 2. Identifying the Distinctions between S-MPC and AR-RNN-LSTM

Optimal control determines the actions that optimize a performance objective by solving a sequential decision-making problem. The preceding section highlighted the need to compare S-MPC and AR-LSTM, the two primary approaches for optimal control applied to EM for hybrid MG control, and the possibility of combining them. Both methods utilize certain components, while others are more controller-specific. These formulation differences make it difficult to compare and combine the two approaches, necessitating a conceptual analysis. This assists in identifying the primary methods of optimal control and establishes a common ground for a comprehensive classification. The sections that follow detail the most important aspects of these control methods.

### 2.1. Strategy

There are typically two ways to approach an optimal control problem: by employing the S-MPC-inherent receding horizon principle or formalizing the problem as an AR-LSTM.

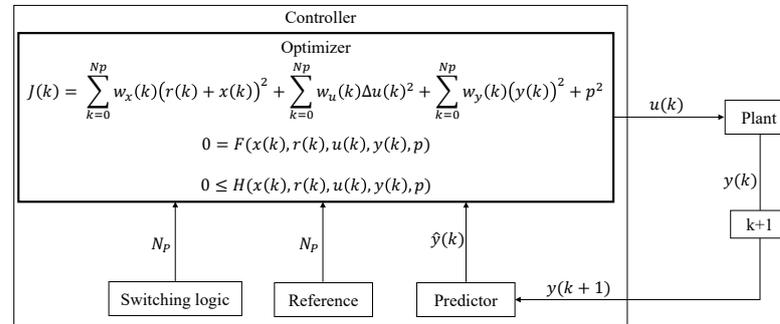
S-MPC is a control strategy that involves using a mathematical model of the system being controlled to predict the system's future behavior and optimize a control signal over a finite time horizon. At each time step, the control signal is updated based on the current state of the system and the predictions made by the model. This is widely used in industrial control applications, such as process control, automotive control, and robotics, where it is important to consider the system dynamics being controlled and to optimize performance over a prediction horizon. At each time step  $k$  in S-MPC, switching logic controls multiple modes for the accumulators, which fully describe the controller model at the current time. Then, the trajectories of the future state  $x$  and input  $u$  are optimized for a prediction horizon  $N_p$  based on the explicit representation of an objective function  $J$  and a controller model  $F$ .  $J$  is the minimization of the imported energy and maximization of the exported energy. The constraints  $H$  are also introduced explicitly in the optimization problem. The objective function, model, and constraints may also depend on the model outputs  $y$  and time-invariant parameters  $p$ . In addition,  $r(k)$  is the reference variable representing the PV, load data, and zero along the prediction horizon  $N_p$ .  $w_x(k)$  and  $w_u(k)$  are weighting coefficients reflecting the relative significance of  $x(k)$  and penalizing relatively large variations in  $u(k)$ , respectively. Only the initial control input from the optimized trajectory [16] is Implemented. Figure 1a depicts the full S-MPC procedure.

$$J(k) = \sum_{k=0}^{N_p} w_x(k)(r(k) + x(k))^2 + \sum_{k=0}^{N_p} w_u(k)\Delta u(k)^2 + \sum_{k=0}^{N_p} w_y(k)y(k)^2 + p^2 \quad (1)$$

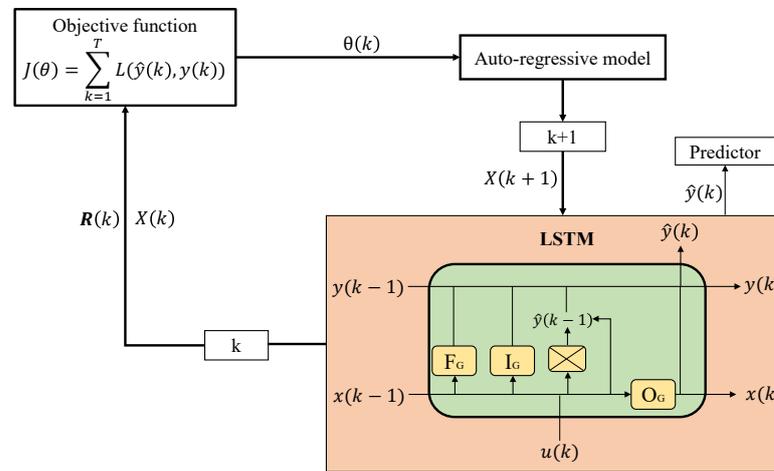
In the application of S-MPC to EM of MGs, the state vector  $x$  represents the state of charge of the accumulators ( $SOAcc$ ), such as the battery, fuel tank, and water tank, and the model output  $y$  illustrates the imported and exported energy, such as a grid to the load  $GR_{LD}$  and PV to the grid, and the battery ( $PV_{GR} + PV_{BAT}$ ). Depending on whether or not the controller model employs physical insights, the set of time-invariant parameters  $p$  may or may not represent the physical properties of the MG.

In contrast to RNN-LSTM, AR models are not neural network architectures. On the contrary, they are statistical models that identify dependencies and patterns within a time series, based on its own lagged values. The AR model predicts the future values of a variable based on its historical values and the estimated coefficients during model training. In other words, AR models are a statistical modeling technique that assumes a variable's current value is a function of its previous values. They are frequently utilized for time series analysis and forecasting. Therefore, AR models can be viewed as a linear regression, in which the predictors are the values of the same variable at a prior time [32]. AR models can be used to model the system's dynamics within the context of control systems or reinforcement learning. A model can predict future states or observations by estimating the AR coefficients. These predictions can then be fed into control algorithms or reinforcement learning agents, in order to optimize control signals or decision-making.

Unlike neural network architectures, AR models are not adaptive by nature. The estimation of AR coefficients requires training on historical data, and their performance may degrade if the underlying dynamics of the system change significantly over time.



(a) S-MPC



(b) AR-LSTM

**Figure 1.** Block diagram of (a) S-MPC and (b) AR-LSTM.

The following equation can mathematically represent an AR model of order  $q$  [32]:

$$X(k) = c + \varphi(1)X(k - 1) + \varphi(2)X(k - 2) + \dots + \varphi(q)X(k - q) + \varepsilon(k) \quad (2)$$

where  $X(k)$  represents the value of the time series at time  $k$  in this equation.  $c$  is a constant term or an intercept.  $\varphi$  terms represent AR model coefficients. The coefficients or weights associated with the previous values of the time series are denoted by  $1, 2, \dots, q$ .  $X(k - 1)$ , and  $X(k - 2), \dots, X(k - q)$  represent the lagged values of the time series at time points  $k - 1, k - 2, \dots, k - q$ , respectively.  $\varepsilon(k)$  is the error term or random noise at time  $k$ , representing the data portion the model cannot explain.

RNN-LSTM is a neural network type ideally suited to processing sequential data. Unlike feed-forward neural networks, it has loops that allow information to be passed from one sequence step to the next. The approach for employing RNN-LSTM includes selecting an appropriate network architecture, an optimization algorithm for training the network, and an appropriate set of hyperparameters. RNN-LSTM is an extension of a feed-forward neural network with internal memory. RNN-LSTM is recurrent because it performs the same function for each data input, while the output of the current input is dependent on the previous computation. After the output has been generated, it is duplicated and sent back into the recurrent network [69]. For decision-making, it considers both the current input and the output from the previous input it learned. As shown in Figure 1b, the input vector of an LSTM network is  $u(k - 1)$  at time step  $k$ .  $y(k)$  represents the output vectors passed through the network between time steps  $k$  and  $k + 1$ . Three gates update and control the

cell states in an LSTM network: the forget gate, input gate, and output gate. The gates are activated by hyperbolic tangent and sigmoid functions. Given new information that has entered the network, the forget gate determines which cell state information to forget. Given new input information, the input gate determines what new information will be encoded into the cell state. Using the output vector  $y(k)$ , the output gate controls what information encoded in the cell state is sent to the network as input in the subsequent time step.

In the mathematical modeling of RNN-LSTM, the current state can be expressed mathematically as

$$x(k) = f(x(k - 1), u(k)) \tag{3}$$

where  $x(k)$  represents the current state,  $x(k - 1)$  represents the previous state, and  $u(k)$  is the current input. Because the input neuron would have applied the transformations to the previous input, we now have the state of the previous input rather than the input itself. Each successive input is, therefore, referred to as a time step.

Considering the simplest form of RNN-LSTM, where the activation function is tanx, the weight at the recurrent neuron is  $W_{xx}$ , and the weight at the input neuron is  $W_{ux}$ , we can write the equation for the state at time  $k$  as follows [69]:

$$x(k) = \tan x(W_{xx}x(k - 1) + W_{ux}u(k)) \tag{4}$$

In this instance, the recurrent neuron only considers the previous state. The equation may involve multiple such states for longer sequences. After calculating the final state, the output can be generated. Once the current state has been computed, we can then calculate the output state as follows [69]:

$$y(k) = W_{xy}x(k) \tag{5}$$

where  $y(k)$  is the output state and  $W_{xy}$  is the weight at the output state. This process is represented in Figure 2.

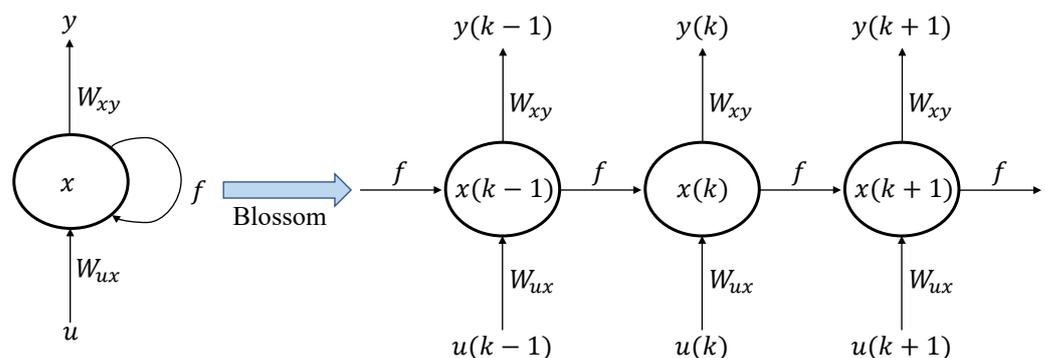


Figure 2. Structure of the RNN.

First, it extracts  $u(0)$  from the input sequence and then outputs  $y(0)$ , which, along with  $u(1)$ , is the input for the subsequent step. Therefore,  $y(0)$  and  $u(1)$  are the inputs for the subsequent step. Similarly,  $y(1)$  from the subsequent step is the input for  $u(2)$  for the subsequent step, and so on. Consequently, it remembers the context throughout training.

A cost function quantifies “how well” a neural network performs with respect to the training sample and the expected output. This may also depend on factors such as weights and biases. This is a single value, not a vector, because it evaluates the overall performance of the neural network. The objective of the cost function is to evaluate the

network's performance, to minimize its value during training. The cost function for a typical RNN-LSTM is the sum of losses at each time step [70].

$$J(\theta) = \sum_{k=1}^T L(\hat{y}(k), y(k)) \quad (6)$$

where  $\theta$  represents the parameters of the RNN,  $T$  represents the length of the input sequence,  $\hat{y}_t$  represents the predicted output, and  $y_t$  represents the actual output at time step  $k$ .  $L$  is the loss function quantifying the difference between the predicted and actual output. The RNN's training parameters are adjusted to minimize the cost function using gradient descent or a comparable optimization algorithm. The objective is to identify the parameters that minimize the loss over all time steps, resulting in an RNN that can accurately predict the output for a given input sequence.

### 2.2. Problem-Solving Method

By analyzing the control processes illustrated in Figure 1a,b, it is possible to identify a number of expressions with total or partial equivalence between the two methods.

S-MPC can be solved implicitly by performing switching logic, forecasting, and resolving a dynamic optimization problem at each time step or explicitly by learning a control policy from data generated by a S-MPC with any function approximation. Consequently, S-MPC has a higher online computational cost, because every control step requires estimation of the states and dynamic optimization. Typically, the optimization problem in S-MPC is solved using numerical optimization techniques, such as nonlinear programming or quadratic programming (QP) (in this paper, QP has been used), to solve the optimization problem. The solution to the optimization problem over the prediction horizon provides the optimal control signal. At each time step, the first component of the optimal control signal is applied to the system, and the process is repeated with the updated state and prediction horizon values. S-MPC requires the solution of an optimization problem at each time step, which can be computationally expensive for large systems.

The training process for AR-LSTM involves back-propagation through time (BPTT), a variation in the back-propagation algorithm that considers temporal dependencies in the data. The RNN is unrolled throughout the training for a predetermined number of time steps, and gradients are calculated at each step. The RNN's weights are then updated based on the gradients accumulated across all time steps. The most prevalent optimization algorithm for training RNNs is gradient descent, which involves updating the weights iteratively in the direction of the loss function's negative gradient [69]. However, the standard gradient descent algorithm is susceptible to issues such as vanishing gradients, in which the gradients become extremely small and the weights do not update. Several variants of gradient descent, such as the adaptive gradient descent algorithms AdaGrad, RMSProp, and Adam, have been developed to address this issue [71].

### 2.3. Peak Performance

In S-MPC, the quality of the optimization solution depends on the controller model's precision, which is frequently simplified for computational purposes. Stability and practicability are intrinsically ensured for S-MPC, whereas there is only an immature theory for these issues in AR-LSTM [72]. The absence of safety guarantees in AR-LSTM results from the constraints not being directly imposed in the formulation of the solution method. The optimality of the S-MPC solution depends on the accuracy of the model used to predict the system's behavior and the optimization algorithm's ability to find the optimization problem's global optimum. If the model is inaccurate or the optimization algorithm fails to find the global optimum, the performance of the S-MPC controller may not be optimal.

The optimality of AR-LSTM relies on several factors, including the network's architecture, the training optimization algorithm, and the complexity of the task being performed. AR-LSTM is capable of achieving high levels of performance on a wide variety of sequen-

tial data processing tasks, such as language modeling, machine translation, and speech recognition. AR-LSTM is able to model complex temporal dependencies in sequential data, which is one of its main advantages. The ability of AR-LSTM to incorporate feedback loops enables them to capture long-term dependencies that would be challenging to represent using other models, such as a gated recurrent unit (GRU). In addition, the ability to incorporate memory into the network via mechanisms improves the performance of RNNs in tasks requiring long-term memory. Nonetheless, several factors can restrict the optimality of AR-LSTM. One difficulty is the issue of vanishing and exploding gradients, which can hinder the network's ability to discover long-term dependencies. This issue can be mitigated by employing specialized units, such as LSTM and GRU, and optimization algorithms designed to deal with these issues. Another issue is overfitting, which can occur when the model becomes excessively complex and begins to fit the noise in the data rather than the underlying patterns. This can be remedied by employing regularization techniques such as early stopping and dropout [73].

#### 2.4. Computational Effort

S-MPC can require significant computational effort, especially for large-scale systems. S-MPC necessitates the solution of an optimization problem in each time step, which can be computationally costly. Moreover, a significant disadvantage of S-MPC is the need to solve an optimization problem online, which can be complex and involve many optimization variables. Consequently, controller models for S-MPC are commonly simplified at the expense of optimality, and gains in optimization solver efficiency are highly desired. Moreover, switching logic and prediction must be performed at each control step. Nonetheless, several techniques have been developed to reduce the computational effort required for S-MPC, such as online optimization and ML techniques that update the optimization problem as the system evolves.

The computational effort required for training and utilizing AR-LSTM can be substantial, especially for large-scale problems with many time steps and/or parameters. BPTT is the primary computational bottleneck, because it is required to compute the gradients of the loss function with respect to the network parameters. Considering that the computational complexity of BPTT scales linearly with the number of time steps, training AR-LSTM on lengthy sequences can be computationally expensive. In addition, the number of network parameters can contribute to the computational complexity, as larger networks require more computation to update weights during training and to make predictions during inference. Several techniques have been developed to ease these computational challenges, including mini-batch training, which involves updating the weights based on a subset of the training data at each iteration, and gradient clipping, which involves capping the magnitude of gradients to prevent gradients from exploding during training [69].

### 3. Switched Auto-Regressive Neural Control (S-ANC)

This section introduces the specifics of the proposed novel S-ANC algorithm. The objective is to learn from the architecture of RNN-LSTM, while satisfying constraints. Switching logic, dynamic optimization, and learning are the elements from the control and ML communities that are effectively combined to achieve this objective. First, Section 3.1 introduces the hybrid MG structure. Section 3.2 provides an overview of how S-MPC and AR-LSTM are merged logically. Then, Section 3.3 formally describes the S-ANC algorithm.

#### 3.1. Hybrid MG Description

This is a case study of a system constructed in Xanthi, Greece [74]. As depicted in Figure 3, the hybrid MG is comprised of a 15 kW PV array; a battery (BAT), a water tank (WT), and a fuel tank (FT) serving as energy storage systems (ESSs); an electrolyzer (EL); and a fuel cell (FC), as well as the utility grid (GR). The PV can be utilized in the hybrid MG as the primary energy source. If the PV cannot provide sufficient power, the BAT or the FC will meet the load. The GR will provide energy if the battery is depleted and no

hydrogen is available. Alternatively, when the BAT is full and there is an excess, the EL will be utilized if there is space in the WT and the FT. The energy will then be sent to the GR.

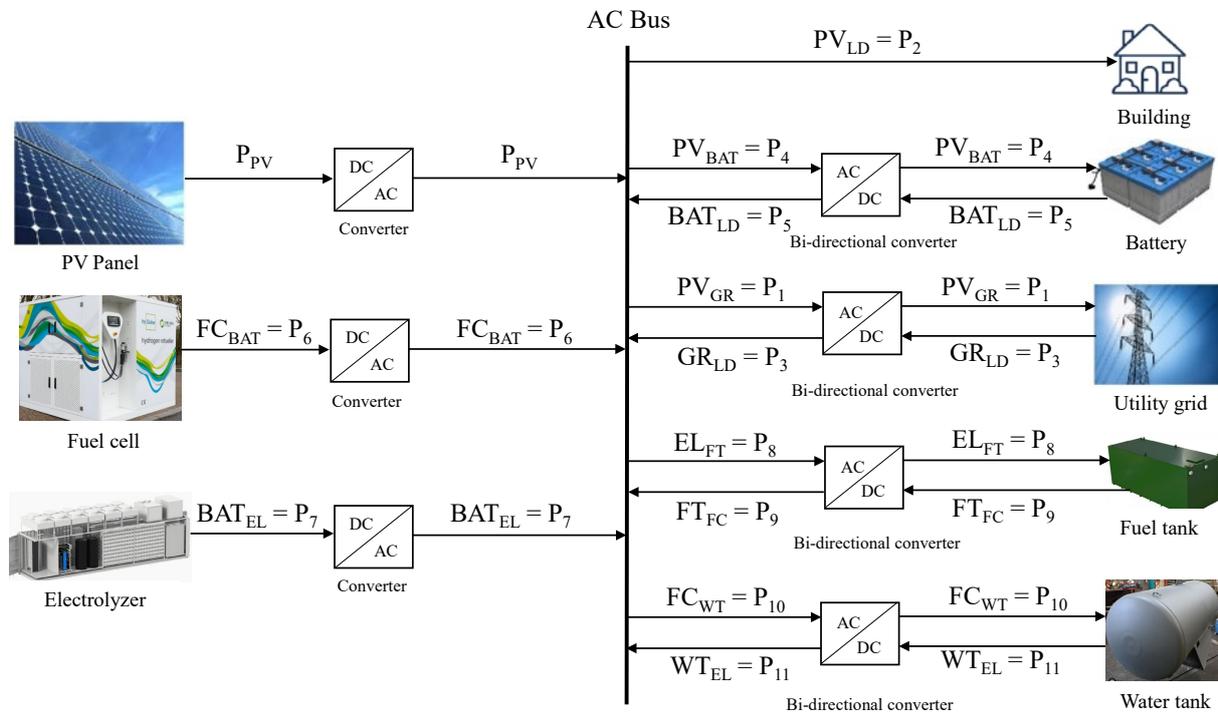


Figure 3. Hybrid MG Structure.

### 3.2. Simple Definition of the Proposed Method

To comprehend the concept underlying S-ANC, one must first comprehend the distinctions between MPC and S-MPC when solving the QP and the two main learning methods of RNN: AR and LSTM. The S-MPC controller must be capable of selecting the appropriate model and control strategy based on the system’s current state, which necessitates additional computational resources and algorithmic complexity. In this paper, for instance, the system dynamics change significantly as the state of each accumulator in the hybrid MG changes; consequently, S-MPC can use different models for various states. This requires creating and validating multiple models, and the S-MPC controller must be able to switch between these models based on the current state.

The construction of S-MPC is challenging and intricate, particularly for a hybrid MG, which must accommodate many operating modes and complex switching conditions. This complexity is caused by a number of factors, including

- **Model development:** S-MPC requires the creation of multiple models that represent the system’s behavior in different operating modes. This requires an efficient system architecture and behavior;
- **Mode detection:** The S-MPC controller must be able to detect the current mode of operation of the system, which can be difficult in certain circumstances;
- **Switching logic:** The S-MPC controller must select the appropriate model and control strategy based on the current operating mode and desired performance objectives. This necessitates the designing of switching logic that maps the system’s current state to the appropriate model and control strategy (a mode’s objective function and an operational mode’s objective function may differ).

The S-MPC solution method takes information from the hybrid MG, such as PV and load data and accumulator parameters, including their charging and discharging efficiencies. Then, the input  $u$ , state  $x$ , and output vectors  $y$  are defined. Based on the controller model, the objective function  $J$  is inferred at each control step using this method.

After that, the state vector is converted to an AR model  $X(k)$ , in order to predict the value at the subsequent time step. This is a straightforward concept that can produce accurate forecasts for various time series problems. Nevertheless, the AR model needs a plant model and a prediction horizon, so the computational time of the model is still high. Therefore, the current state  $x$ , input  $u$ , and output vectors  $y$  are updated using the AR-LSTM method.

S-ANC employs time series value-based AR-LSTM to estimate the value of being in a particular output vector  $\hat{y}$ , as determined by S-MPC, with a prediction horizon of only one control step. By doing so, in S-ANC, the S-MPC method is truncated with the predicted output vector  $\hat{y}$  and optimized by the hybrid MG system during  $k$  steps ahead through employing the AR-LSTM method. Consequently, the principal components of the S-MPC, namely the reference, predictor, and switching logic, remain active in S-ANC; however, the time series value function is utilized to shorten the nonlinear program and enable learning. The interaction of S-ANC's primary components is depicted in a diagram in Figure 4. The merging of S-MPC and AR-LSTM in the S-ANC algorithm is intuitively depicted in Figure 4.

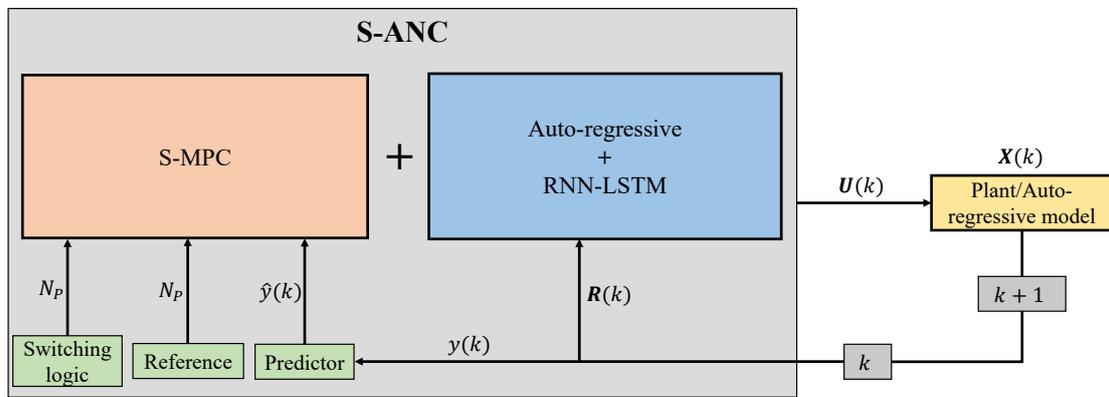


Figure 4. Block diagram showing the introduction of S-ANC.

### 3.3. Formal Definition

Initially, the system state, control, and output vectors are defined for the hybrid MG system in the S-MPC:

The system-state vector of the MG is as follows:

$$x(k) = [SOAcc^l(k)] \tag{7}$$

where  $l \in \{BAT, FT, WT\}$ .  $SOAcc^{BAT}(k)$ ,  $SOAcc^{FT}(k)$ , and  $SOAcc^{WT}(k)$  are the state of the accumulators for the battery, hydrogen tank, and water tank, respectively.

The system-control (input) vector of the MG is defined as follows:

$$u(k) = [P_3(k); P_4(k); \dots P_{11}(k)] \tag{8}$$

The system-output vector of the MG is defined as follows:

$$y(k) = [P_1(k); P_2(k)] \tag{9}$$

Consider the discrete-time linear state-space system:

$$X(k + 1) = A_x x(k) + B_u u(k) \tag{10}$$

where  $k = 0, 1, 2, \dots, N_p - 1$  symbolizes the discrete-time instant.

By defining the following matrices:

$$\mathbf{A}_x = \begin{bmatrix} I \\ A \\ A^2 \\ \vdots \\ A^{N_p} \end{bmatrix} \quad \mathbf{B}_u = \begin{bmatrix} 0 & 0 & \dots & 0 \\ B & 0 & \dots & 0 \\ AB & B & \dots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ A^{N_p-1}B & A^{N_p-2}B & \dots & B \end{bmatrix} \quad (11)$$

where

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad B^T = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \eta_{ch} & 0 & 0 \\ -\eta_{dis} & 0 & 0 \\ \eta_{ch} & 0 & 0 \\ -\eta_{dis} & 0 & 0 \\ 0 & \eta_{ch,H_2} & 0 \\ 0 & -\eta_{dis,H_2} & 0 \\ 0 & 0 & \eta_{ch,H_2O} \\ 0 & 0 & -\eta_{dis,H_2O} \end{bmatrix} \quad (12)$$

The linear state-space equation can be stated depending on the battery, fuel tank, and water tank equations as follows [75]:

$$SOAcc^l(k+1) = SOAcc^l(k) + \frac{P_{a \rightarrow b}^j(k) - P_{b \rightarrow a}^j(k)}{C_l} \quad (13)$$

where  $j$  is the energy flows, so  $j \in \{Power, Hydrogen, Water\}$ .  $a \rightarrow b$  represents the energy flows between accumulators and converters; for example,  $P_4$  is the power from the PV to the battery.

Define the constraints for the hybrid MG: Energy flows from the PV, GR, BAT, FT, EL, FC, and WT are positive and subject to maximum values.

$$\begin{aligned} 0 \leq P_1(k) = P_{LD}(k) - y_a(k) &\leq P_1^{max} \\ 0 \leq P_m^j(k) &\leq P_m^{jmax} \end{aligned} \quad (14)$$

where  $P_m^{jmax}$  ( $m = 1, 2, \dots, 11$ ) imply the maximum values of energy/matter flows.

The sum of PV energy supplied directly for the load ( $P_2(k)$ ) and the battery for the charging ( $P_4(k)$ ) should be smaller than the energy flow from the PV array, ( $P_{PV}(k)$ ).

$$P_2(k) + P_4(k) \leq P_{PV}(k) \quad (15)$$

The  $SOAcc^l$  is restricted between the minimum and maximum values [16].

$$SOAcc^{lmin} \leq SOAcc^l \leq SOAcc^{lmax} \quad (16)$$

Define the reference matrix ( $\mathbf{R}$ ) for the hybrid MG system:

$$\mathbf{R}(k) = [w_{xx}(P_{LD}(k); P_{PV}(k); 0); \dots; P_{LD}(k + N_p - 1); P_{PV}(k + N_p - 1); 0] \quad (17)$$

Design and control the multiple models (converting MPC to S-MPC) depending on several parameters as follows:

$$\text{if } P_{a \rightarrow b}^j < 0; \quad SOAcc^l > SOAcc^{lmin} \quad \text{and} \quad SOAcc^l < SOAcc^{lmax} \quad (18)$$

$$\text{if } P_{a \rightarrow b}^j > 0; \quad SOAcc^l > SOAcc^{lmax} \quad \text{and} \quad SOAcc^l < SOAcc^{lmin} \quad (19)$$

where  $i = 1, 2, \dots, 11$ .

Regarding the AR-LSTM formulation, if Equations (2) and (4) are merged, the new state vector will be as follows:

$$\begin{aligned} \mathbf{X}(k) = & c + \varphi(1)\tan x(W_{xx}x(k-1) + W_{ux}u(k)) + \varphi(2)\tan x(W_{xx}x(k-2) \\ & + W_{ux}u(k-1)) + \dots + \varphi(q)\tan x(W_{xx}x(k-N_p) + W_{ux}u(k-N_p+1)) + \varepsilon(k) \end{aligned} \quad (20)$$

The objective function of the hybrid MG system using S-ANC (the combination of Equations (1) and (6)):

$$\begin{aligned} J(k) = & \sum_{k=0}^{N_p} W_{xx}(k)(\mathbf{R}(k) + \mathbf{X}(k))^2 + \sum_{k=0}^{N_p} W_{ux}(k)\Delta u(k)^2 \\ & + \sum_{k=0}^{N_p} W_{yy}(k)y(k)^2 + p^2 + \sum_{k=0}^{N_p} \hat{y}(k), y(k) \end{aligned} \quad (21)$$

$$\begin{aligned} 0 = & F(\mathbf{X}(k), y(k), u(k), \hat{y}(k), \mathbf{R}(k), p) \\ 0 \leq & H(\mathbf{X}(k), y(k), u(k), \hat{y}(k), \mathbf{R}(k), p) \end{aligned} \quad (22)$$

The main advantage of employing the formulation presented by Equations (21) and (22) is that it imposes short-term safety constraints, while allowing for continuous empirical experience-based learning. In addition, reducing the prediction horizon of the dynamic optimization problem significantly simplifies the resulting nonlinear program. Notably, both optimization functions from Equation (21) must be jointly merged, such that the state  $\mathbf{X}$  must be related to the expected optimization variables in  $k + 1$ . This results in lower overheads than optimizing with longer prediction horizons that must be discretized over time.

Notably, domain knowledge is encoded in the controller model  $F$  for optimization and control vectors, providing the algorithm with understandability. Then, the constraints are implied for the hybrid MG system. The next step is to automatically convert the traditional MPC into S-MPC. The final steps in the S-MPC are to solve the cost function and obtain “optimal decision variables”, as shown in Algorithm 1. After that, the hybrid AR-LSTM method is initiated by configuring the controller model  $F$ . The current state  $\mathbf{X}$  is found using Equation (20), before training the “optimal control decisions”. Finally, the control variable  $\mathbf{U}$  and  $\hat{y}$  are solved by utilizing updated reference  $\mathbf{R}$  and Equations (21) and (22).

To begin, design a model of the MG system. The system reads some MG specifications, such as PV and load data, accumulator data, and the maximum values of power flows among the components of the hybrid MG. Following that, the MPC controller is implemented, which will state the optimization problem and solve it at each time step, to obtain the optimal control inputs for the next time step. However, the MPC is converted into the S-MPC before it is applied. The optimization problem should consider the objectives and constraints given in this paper’s methodology section. Implement an AR-LSTM model and train it on past data to increase the accuracy of the predictive model utilized by the S-MPC controller. Based on the present and previous system conditions, the AR-LSTM should be able to anticipate future MG behavior. The prediction should be input into the S-MPC controller’s optimization problem. Finally, as indicated in the methodology section of this

paper, the S-MPC and AR-LSTM controllers in a closed-loop control system are combined. The proposed method can test the control strategy under various operating situations and evaluate its performance using the provided performance criteria (cost functions).

---

**Algorithm 1:** Switched Auto-regressive Neural Control (S-ANC)

---

**Identify:**  $F(x(k), r(k), u(k), y(k), p) = 0$   
**Imply:**  $H(x(k), r(k), u(k), y(k), p) \geq 0$   
**Switching logic:** Conversion MPC into S-MPC  
**Solve:** Objective function for S-MPC using Equation (1)  
**Obtain:** “Optimal decision variables”  
**Configure:**  $F(\mathbf{X}(k), y(k), u(k), \hat{y}(k), \mathbf{R}(k), p) = 0$   
**while true do**  
     $\mathbf{X}(k) \leftarrow$  current state using Equation (13)—AR-LSTM  
    Pretrain “optimal decision variables” using  $F$   
     $\mathbf{U}(k) \leftarrow$  control (input) variable  
     $\mathbf{R}(k) \leftarrow$  estimate from measurements  
     $\hat{y}(k) \leftarrow$  Solve Equations (14) and (15)  
     $\mathbf{R}(k+1), \mathbf{U}(k+1) \leftarrow$  Apply  $\hat{y}(k)$  to  $F$   
**end**

---

More specifically, to implement our proposed method in operation, initially, model the MG system and the S-MPC and AR-LSTM controllers, and then combine these models into a closed-loop control system. Here are the detailed steps that need to be followed, as illustrated in Figure 5:

- Initiate the system specifications and operational conditions from the MG operator;
- Solve the systematic generation of the control problem, employing the MPC with the QP;
- Using switching logic, automatically convert the MPC into the S-MPC;
- The optimal control decisions are obtained;
- The optimal control decisions are employed as input data for the AR method;
- The data preparation is initiated. This step has several parameters, such as data cleaning, extracting features, and merging the input data and PV constraints;
- The AR model is implemented to increase the accuracy of our proposed method;
- After that, multivariate time series are employed;
- Then, the training and test data are selected and evaluated;
- To move the LSTM layer to after the RNN, a sequential network of an input LSTM layer is produced;
- In this step (implementation of LSTM), several parameters are defined, including the batch size, epoch number, and type of optimizer;
- Before moving the calculation to the model accuracy, the scaling for the forecast and actual data are inverted;
- The model accuracy is calculated using various methods, along with the mean directional accuracy,  $R^2$  method, and so on;
- Integrate the S-MPC and AR-LSTM controllers into a closed-loop control system by connecting the RNN output to the MPC controller’s input, and the MPC controller’s output to the MG system’s input;
- Then, the optimal control decisions and references are updated. In other words,  $\mathbf{X}$ ,  $\mathbf{U}$ , and  $\mathbf{R}$  are re-evaluated depending on the model accuracy;
- If this accuracy is unreasonable, the S-MPC is reapplied using the updated control decisions.

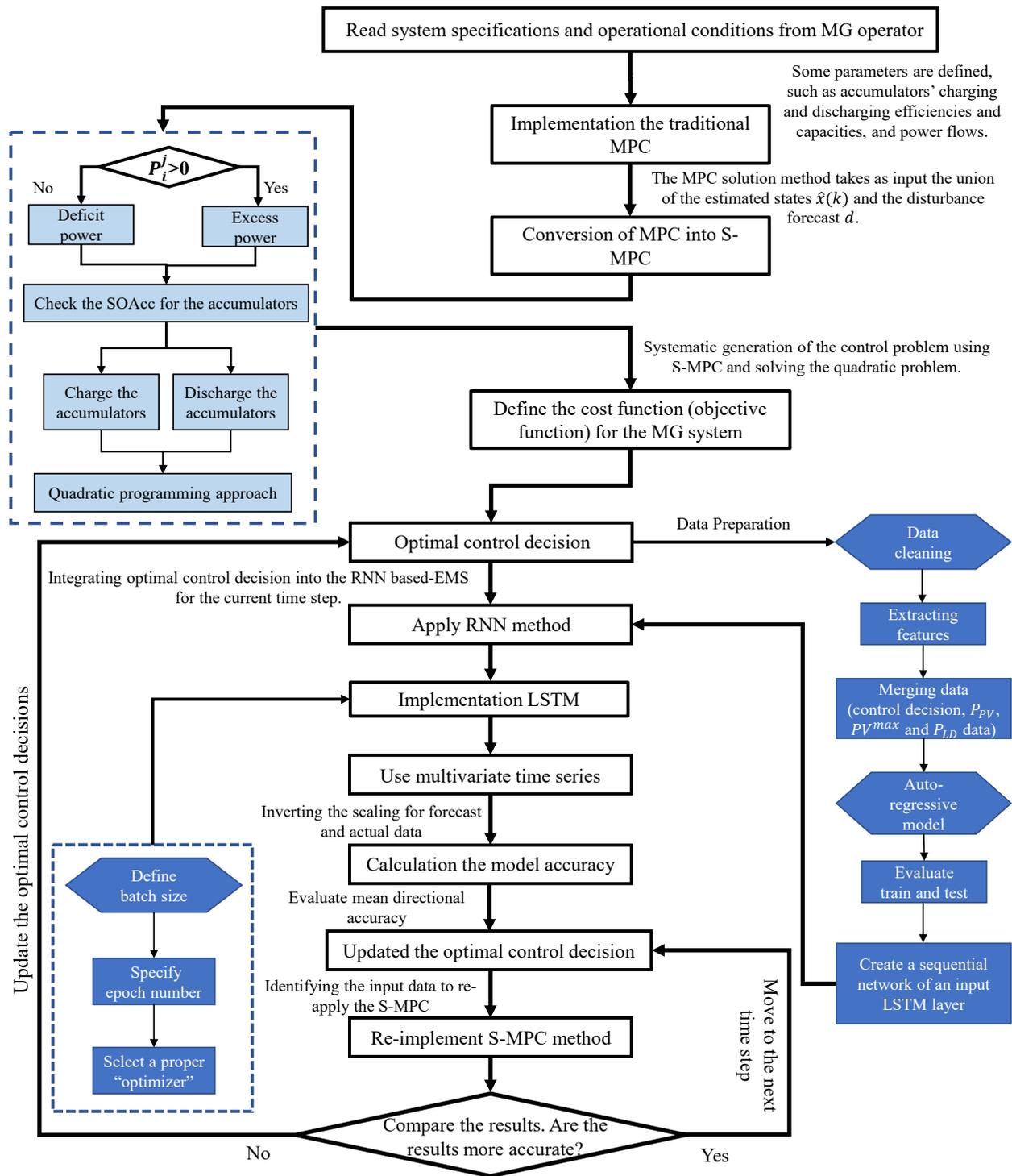


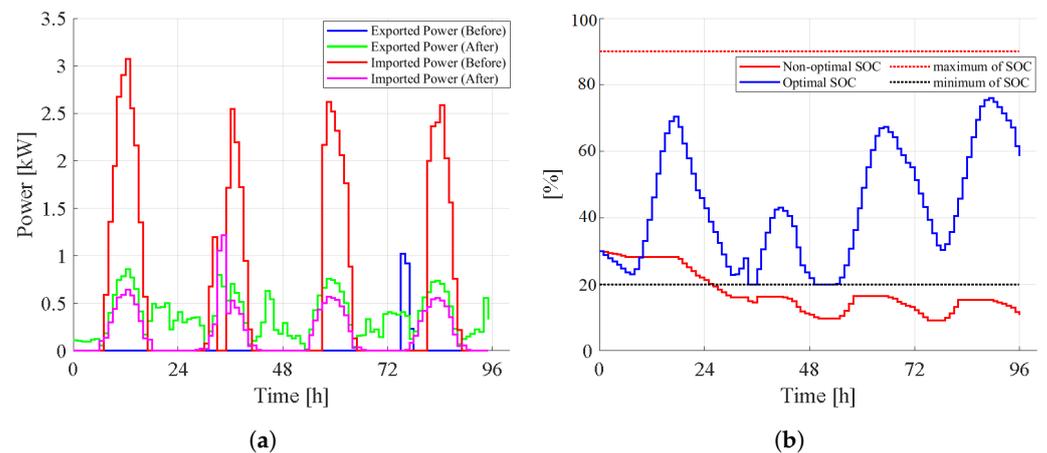
Figure 5. Flow chart of the proposed method.

#### 4. Results and Discussion

##### 4.1. Case 1: Implementation of S-MPC

The non-optimal and optimal control (S-MPC) were compared for 96 h (four days) in this case. In other words, in case 1, the emphasis was on the optimization of S-MPC and its effect on the power flows of the system (Figure 6a) and the SOC of the battery (Figure 6b). Non-control methods employ simplistic control strategies or heuristic rules, disregarding the system's dynamic nature. In addition, they do not have any constraints, so there are disadvantages associated with this strategy, such as poor SOC management,

potential deviations from desired SOC levels, and an inability to adapt to changing system conditions. As shown in Figure 6b, the SOC of the battery went below the critical value (20%), since the non-optimal method had no constraints. In contrast, the S-MPC method is an alternative to the non-optimal control method. S-MPC's ability to dynamically select the appropriate model and control strategy based on the system's current state is one of its key advantages. S-MPC optimizes control actions to achieve the desired performance objectives, particularly in effectively managing the SOC of a battery.



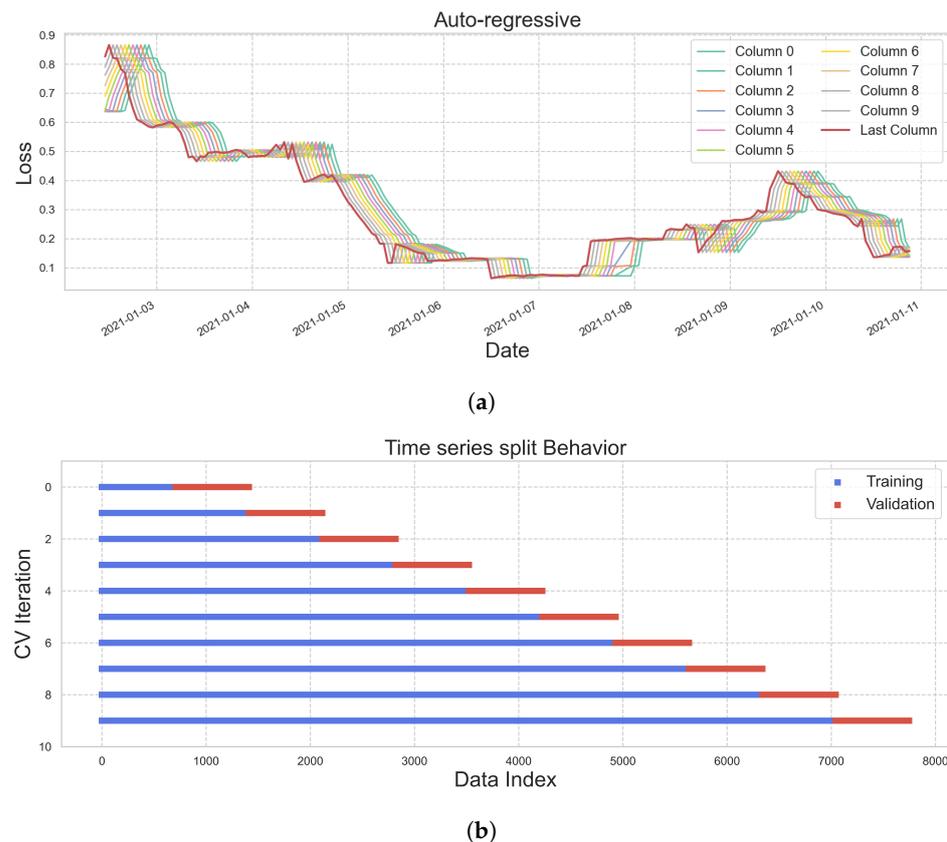
**Figure 6.** Comparison of (a) power flows and (b) SOC of the battery using the optimal method (S-MPC) and non-optimal method.

The S-MPC controller was designed to select the optimal model and control strategy based on the current operating mode and performance objectives. S-MPC enables enhanced power flow control and EM by effectively adapting to the changing dynamics of the hybrid MG, utilizing distinct models for each state. Case 1's implementation of the S-MPC controller successfully optimized the hybrid MG system's power flows. It substantially reduced energy imports and increased energy exports, resulting in a more efficient use of resources and enhanced energy flow management. The improved control strategy enabled the MG to operate closer to its optimal performance, enhancing its dependability and reducing the operational costs. However, it is important to note that developing and implementing the S-MPC controller for the hybrid MG system presented obstacles due to the complex switching conditions and multiple operating modes. To guarantee the selection of the optimal model and control strategy, the switching logic had to be meticulously designed. The controller's increased complexity required more computational resources than traditional MPC methods. The model's computational time was almost 405 s.

#### 4.2. Case 2: Implementation of the Merged S-MPC and AR

Case 2 investigated the integration of the AR model with S-MPC. AR models accurately predict future time steps by capturing the time series behavior of the system. Various analyses, including variations, cross-validation (CV) iteration-time series behavior for training and validation, CV iteration-training data for each CV iteration, and predictions ordered by test prediction number were used to evaluate the performance of the AR models.

Figure 7a shows a visualization designed to provide insight into the behavior of the lagged target feature over time. We can identify patterns, trends, and correlations within the lagged target data by examining the plot. Understanding the characteristics of the lagged target can aid in developing and optimizing an AR linear regression model that uses this characteristic for prediction. By displaying the lagging feature of the target, we can observe its values across multiple time steps. This lets us determine whether the lagged target exhibits specific patterns, trends, or variations over time.



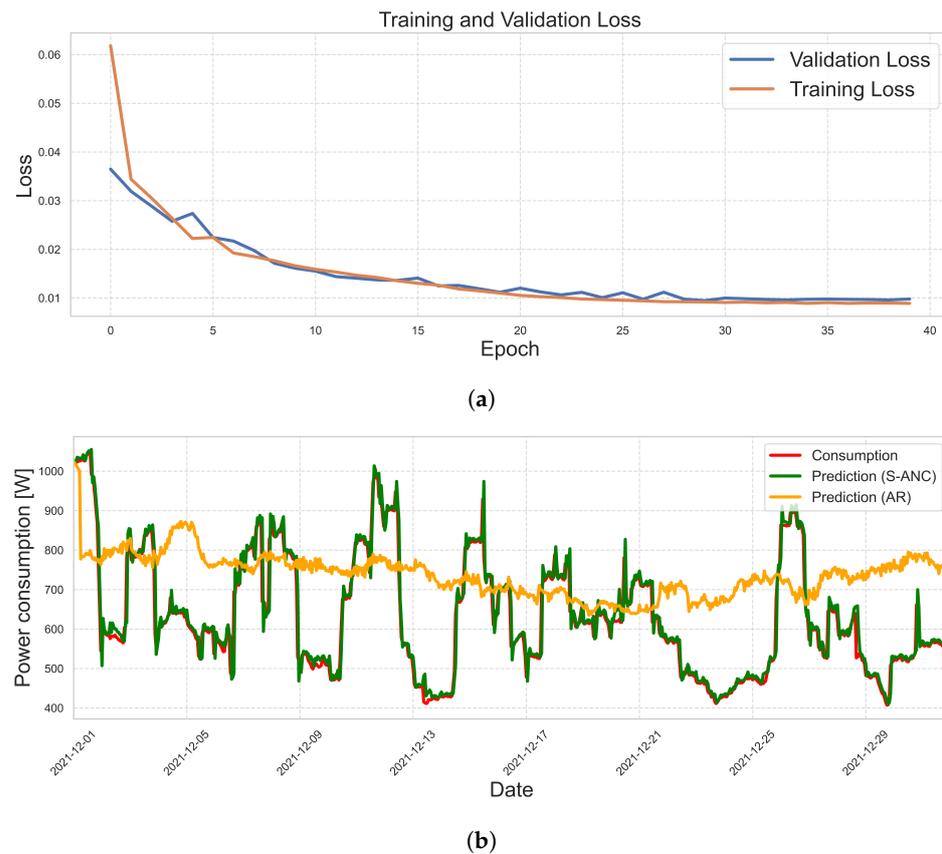
**Figure 7.** Visualization of (a) the behavior of the lagged target feature over time, (b) the adaptability of the AR models to various patterns and tendencies

As illustrated in Figure 7b, the variations in the AR models illustrate their capacity to capture and model the system's complex dynamics. By analyzing the CV iteration-time series behavior, the adaptability of the AR models to various patterns and tendencies in the training and validation datasets was assessed. This analysis shed light on how the models learn and generalize from the available data, enabling accurate predictions for various time series problems.

Case 2's successful integration of AR models with S-MPC illustrated the importance of incorporating time series behavior and forecasting capabilities into a control system. Combining S-MPC and AR models permitted enhanced adaptation to system dynamics and improved the prediction accuracy, thereby enhancing the MG's overall control performance.

#### 4.3. Case 3: Implementation of the S-ANC

Case 3 examined the combination of S-MPC and AR-LSTM models. Integrating these advanced models aimed to enhance the predictive capabilities of the control system. The S-ANC predicted the last month of the year using the first eleven months as training data. Using metrics such as train-test (Figure 8a) and grid consumption prediction with AR regression and S-ANC (Figure 8b), the performance of this merged approach was evaluated. This integration (S-MPC and AR-LSTM) increased the precision of forecasting and the precision of power flow optimization. The enhanced prediction capabilities of the AR-LSTM models allowed the control system to anticipate future energy requirements and adjust the operation of the MG accordingly. In Case 3, the S-MPC controller was improved by combining it with AR and RNN-LSTM models. This integration aimed to improve the precision of predictions and the overall performance of the control system. Two primary figures were generated for analysis: a comparison between the training and the test, and a prediction of grid consumption using AR regression and S-ANC.



**Figure 8.** Illustration of (a) train–test data and (b) prediction of grid consumption using AR regression and S-ANC.

The train–test comparison diagram visually represents the AR-LSTM models' capacity to generalize effectively for unobserved data. It compares the predicted grid consumption values during the testing phase with the actual values, indicating the AR-LSTM models' ability to capture the hybrid MG's complex patterns and dynamics (Figure 8b). The diagram depicts the performance of the combined S-MPC, AR, and RNN-LSTM models on the training and testing datasets (Figure 8a). According to our simulation, the model could generalize well for new data and the integration strategy's effectiveness. Moreover, the prediction of grid consumption using AR regression and S-ANC illustrated the ability of the combined method to optimize power flows, while accurately predicting future power demands. By leveraging the predictive capabilities of AR-LSTM models within the S-MPC framework, the control system could more precisely estimate grid consumption, allowing for more effective EM and enhancing the MG's adaptability to load demands and renewable energy generation fluctuations. The grid consumption forecast graph (Figure 8b) depicts the projections generated by the AR and S-ANC models. This enables a comparison of the two methods and highlights the advantages of the S-ANC method, which employs the AR-LSTM model for accurate predictions, while reducing computational time. The computational time of the model was reduced by nearly 214 s.

#### 4.4. Calculation of Model Accuracy

To comprehensively evaluate the performance of the S-ANC prediction model, three unified evaluation indices, including R-squared score, mean absolute error (MAE), and mean square error (MSE) [76,77] were selected in this paper. The evaluation indexes have the following mathematical definitions:

**R-squared Score:**

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (23)$$

**Mean Absolute Error:**

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n} \quad (24)$$

**Mean Squared Error:**

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (25)$$

where  $y_i$  is the output vector using the S-MPC;  $\hat{y}_i$  represents the predicted value of the output vector by employing the S-ANC;  $\bar{y}$  represents the average value of output vector;  $n$  represents the total number of samples.

In this study, we assessed the performance of our predictive model using various accuracy metrics. The R-squared error, which measures the proportion of the variance in the dependent variable that can be predicted from the independent variables, was 0.951. This suggests that our model accounted for approximately 95.1% of the data variance, indicating a strong predictive ability. In addition, we determined that the MAE was 0.571. The MAE is the mean absolute difference between observed (from the MPC) and predicted values. A smaller MAE indicates that the predicted and observed values correspond more closely. In our case, the relatively low MAE indicates that our model's predictions deviated from the true values, on average, by approximately 0.571%. Likewise, we determined the MSE to be 0.571. The MSE measures the average squared deviation between predicted and observed values. Similarly to the MAE, a lower MSE indicates greater precision. The MSE value of our model indicated that the squared differences between the predicted and observed values were, on average, 0.571 units. Overall, the results show that our predictive model was effective. The relatively low MAE and MSE values of 0.571 indicated precise predictions, with minimal deviations from the actual values.

## 5. Conclusions

Our findings show the efficacy and advantages of the S-ANC method for the intelligent control and management of hybrid MGs. The optimization of S-MPC improves energy management and power flow control, resulting in more efficient use of resources. The integration of AR and RNN-LSTM models improves the accuracy of predictions, allowing the control system to adapt to dynamic system conditions and to optimize the operation of the MG. The successful implementation of S-ANC significantly affects the dependability, sustainability, and cost-effectiveness of hybrid MG systems. We can achieve efficient control and management of complex energy systems by leveraging the capabilities of advanced modeling techniques within the S-MPC framework. These findings support the incorporation of hybrid MGs in future energy systems and contribute to developing intelligent control strategies. By combining the AR-LSTM, the computational time of the model was reduced by approximately 47.2%. In addition, this study assessed the accuracy of our predictive model. The R-squared error, which quantifies the amount of variance in the dependent variable that can be predicted from the independent variables, was 0.951. Our model predicted 95.1% of the variance in the data, indicating a high level of predictive ability. The MAE and MSE values of 0.571 indicated precise forecasts, with minimal deviations from the actual values. The focus of future research and development should be on validating larger-scale systems and incorporating additional advanced models. These developments will enhance the performance and applicability of the S-ANC methodology and contribute to the efficient operation and integration of hybrid MGs in future energy systems.

**Author Contributions:** Conceptualization, M.C. and Y.F.U.; methodology, M.C., A.A. and D.G.; software, M.C. and A.A.; validation, M.C., H.A. and A.A.; investigation, M.C.; writing—original draft preparation, M.C. and Y.F.U.; writing—review and editing, M.C., Y.F.U., H.A., A.A., K.A. and D.G.; visualization, M.C., Y.F.U. and H.A.; supervision, A.A., K.A. and D.G. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
AR	Auto-regressive
AR-LSTM	Auto-regressive Long Short-Term Memory
ARIMA	Auto-regressive Integrated Moving Average
ARMA	Auto-regressive Moving Average
BAT	Battery
BPTT	Back-Propagation Through Time
CNN	Convolutional Neural Network
CV	Cross-validation
DLC	Direct Load Control
EL	Electrolyzer
EM	Energy Management
ESS	Energy Storage System
FC	Fuel Cell
FT	Fuel Tank
GHI	Global Horizontal Irradiance
GR	Grid
GRU	Gated Recurrent Unit
IoT	Internet of Things
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MG	Microgrid
MSE	Mean Squared Error
MILP	Mixed Integer Linear Programming
ML	Machine Learning
MPC	Model Predictive Control
NARX	Nonlinear Auto-regressive with exogenous input
NMG	Networked Microgrid
RNN	Recurrent Neural Network
RES	Renewable Energy Source
S-ANC	Switched Auto-regressive Neural Control
S-MPC	Switched Model Predictive Control
SVM	Support Vector Machine
QP	Quadratic Programming
WT	Water Tank
$\eta_{ch}^l$	Charging efficiency of accumulator $l$
$\eta_{dis}^l$	Discharging efficiency of accumulator $l$
$F$	controller model
$H$	constraint
$J$	objective function
$P_m^{max}$	Maximum values of power flows, 5 kW
PV	Photovoltaic
$PV_{GR}$ or $P_1$	Power flow from PV to grid
$PV_{LD}$ or $P_2$	Power flow from PV to load
$GR_{LD}$ or $P_3$	Power flow from grid to load
$PV_{BAT}$ or $P_4$	Power flow from PV to battery
$BAT_{LD}$ or $P_5$	Power flow from battery to load
$FC_{BAT}$ or $P_6$	Power flow from fuel cell to battery
$BAT_{EL}$ or $P_7$	Power flow from battery to electrolyzer
$EL_{FT}$ or $P_8$	Hydrogen flow from electrolyzer to fuel tank

$FT_{FC}$ or $P_9$	Hydrogen flow from fuel tank to fuel cell
$FC_{WT}$ or $P_{10}$	Water flow from fuel cell to water tank
$WT_{EL}$ or $P_{11}$	Water flow from water tank to electrolyzer
$F_{a \rightarrow b}^j(k)$	Flow of $j$ from node $a$ to node $b$
$C_l$	Capacities of accumulator $l$ , [kWh]
$P_{a \rightarrow b}^j$	Power of $j$ from node $a$ to node $b$
$\varphi$	Auto-regressive model coefficient
$N_p$	Prediction horizon, 24 h
$SOAcc^l$	State of accumulator $l$
$SOAcc_{max}^l$	Maximum value state of accumulator $l$
$SOAcc_{min}^l$	Minimum value state of accumulator $l$
$\varepsilon(k)$	error term or random noise at time $k$

## References

- Kumar, A.S.; Ahmad, Z. Model predictive control (MPC) and its current issues in chemical engineering. *Chem. Eng. Commun.* **2012**, *199*, 472–511.
- Garcia, C.E.; Prett, D.M.; Morari, M. Model predictive control: Theory and practice—A survey. *Automatica* **1989**, *25*, 335–348.
- Pamulapati, T.; Cavus, M.; Odigwe, I.; Allahham, A.; Walker, S.; Giaouris, D. A Review of Microgrid Energy Management Strategies from the Energy Trilemma Perspective. *Energies* **2022**, *16*, 289.
- Khawaja, Y.; Qiqieh, I.; Alzubi, J.; Alzubi, O.; Allahham, A.; Giaouris, D. Design of cost-based sizing and energy management framework for standalone microgrid using reinforcement learning. *Sol. Energy* **2023**, *251*, 249–260.
- Nikkhah, S.; Allahham, A.; Bialek, J.W.; Walker, S.L.; Giaouris, D.; Papadopoulou, S. Active participation of buildings in the energy networks: Dynamic/operational models and control challenges. *Energies* **2021**, *14*, 7220. [[CrossRef](#)]
- Parisio, A.; Rikos, E.; Glielmo, L. A model predictive control approach to microgrid operation optimization. *IEEE Trans. Control. Syst. Technol.* **2014**, *22*, 1813–1827.
- Nikkhah, S.; Allahham, A.; Royapoor, M.; Bialek, J.W.; Giaouris, D. A Community-Based Building-to-Building Strategy for Multi-Objective Energy Management of Residential Microgrids. In Proceedings of the 2021 12th International Renewable Engineering Conference (IREC), Amman, Jordan, 14–15 April 2021; pp. 1–6.
- Allahham, A.; Greenwood, D.; Patsios, C.; Taylor, P. Adaptive receding horizon control for battery energy storage management with age-and-operation-dependent efficiency and degradation. *Electr. Power Syst. Res.* **2022**, *209*, 107936.
- Nikkhah, S.; Sarantakos, I.; Zografou-Barredo, N.M.; Rabiee, A.; Allahham, A.; Giaouris, D. A joint risk-and security-constrained control framework for real-time energy scheduling of islanded microgrids. *IEEE Trans. Smart Grid* **2022**, *13*, 3354–3368.
- Ulutas, A.; Altas, I.H.; Onen, A.; Ustun, T.S. Neuro-fuzzy-based model predictive energy management for grid connected microgrids. *Electronics* **2020**, *9*, 900.
- Silvente, J.; Kopanos, G.M.; Dua, V.; Papageorgiou, L.G. A rolling horizon approach for optimal management of microgrids under stochastic uncertainty. *Chem. Eng. Res. Des.* **2018**, *131*, 293–317.
- Parisio, A.; Rikos, E.; Glielmo, L. Stochastic model predictive control for economic/environmental operation management of microgrids: An experimental case study. *J. Process Control* **2016**, *43*, 24–37. [[CrossRef](#)]
- Garcia-Torres, F.; Bordons, C. Optimal economical schedule of hydrogen-based microgrids with hybrid storage using model predictive control. *IEEE Trans. Ind. Electron.* **2015**, *62*, 5195–5207. [[CrossRef](#)]
- Jayachandran, M.; Ravi, G. Decentralized model predictive hierarchical control strategy for islanded AC microgrids. *Electr. Power Syst. Res.* **2019**, *170*, 92–100. [[CrossRef](#)]
- Cavus, M.; Allahham, A.; Adhikari, K.; Zangiabadi, M.; Giaouris, D. Control of microgrids using an enhanced Model Predictive Controller. In Proceedings of the 11th International Conference on Power Electronics, Machines and Drives (PEMD 2022), Newcastle, UK, 21–23 June 2022.
- Cavus, M.; Allahham, A.; Adhikari, K.; Zangiabadi, M.; Giaouris, D. Energy Management of Grid-Connected Microgrids using an Optimal Systems Approach. *IEEE Access* **2023**, *11*, 9907–9919. [[CrossRef](#)]
- Zhu, B.; Tazvinga, H.; Xia, X. Switched model predictive control for energy dispatching of a photovoltaic-diesel-battery hybrid power system. *IEEE Trans. Control. Syst. Technol.* **2014**, *23*, 1229–1236.
- Maślak, G.; Orłowski, P. Microgrid operation optimization using hybrid system modeling and switched model predictive control. *Energies* **2022**, *15*, 833. [[CrossRef](#)]
- Moness, M.; Moustafa, A.M. Real-time switched model predictive control for a cyber-physical wind turbine emulator. *IEEE Trans. Ind. Inform.* **2019**, *16*, 3807–3817. [[CrossRef](#)]
- Pervez, M.; Kamal, T.; Fernández-Ramírez, L.M. A novel switched model predictive control of wind turbines using artificial neural network-Markov chains prediction with load mitigation. *Ain Shams Eng. J.* **2022**, *13*, 101577. [[CrossRef](#)]
- Magni, L.; Scattolini, R.; Tanelli, M. Switched model predictive control for performance enhancement. *Int. J. Control* **2008**, *81*, 1859–1869. [[CrossRef](#)]
- Aguilera, R.P.; Lezana, P.; Quevedo, D.E. Switched model predictive control for improved transient and steady-state performance. *IEEE Trans. Ind. Inform.* **2015**, *11*, 968–977. [[CrossRef](#)]

23. Kwadzogah, R.; Zhou, M.; Li, S. Model predictive control for HVAC systems—A review. In Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering (CASE), Madison, WI, USA, 17–20 August 2013; pp. 442–447.
24. Forbes, M.G.; Patwardhan, R.S.; Hamadah, H.; Gopaluni, R.B. Model predictive control in industry: Challenges and opportunities. *IFAC-PapersOnLine* **2015**, *48*, 531–538. [[CrossRef](#)]
25. Teimourzadeh, S.; Tor, O.B.; Cebeci, M.E.; Bara, A.; Oprea, S.V. A three-stage approach for resilience-constrained scheduling of networked microgrids. *J. Mod. Power Syst. Clean Energy* **2019**, *7*, 705–715. [[CrossRef](#)]
26. Giaouris, D.; Papadopoulos, A.I.; Seferlis, P.; Papadopoulou, S.; Voutetakis, S.; Stergiopoulos, F.; Elmasides, C. Optimum energy management in smart grids based on power pinch analysis. *Chem. Eng.* **2014**, *39*, 55–60.
27. Giaouris, D.; Papadopoulos, A.I.; Voutetakis, S.; Papadopoulou, S.; Seferlis, P. A power grand composite curves approach for analysis and adaptive operation of renewable energy smart grids. *Clean Technol. Environ. Policy* **2015**, *17*, 1171–1193. [[CrossRef](#)]
28. Khanna, S.; Becerra, V.; Allahham, A.; Giaouris, D.; Foster, J.M.; Roberts, K.; Hutchinson, D.; Fawcett, J. Demand response model development for smart households using time of use tariffs and optimal control—The isle of wight energy autonomous community case study. *Energies* **2020**, *13*, 541. [[CrossRef](#)]
29. Oprea, S.V.; Bâra, A. Edge and fog computing using IoT for direct load optimization and control with flexibility services for citizen energy communities. *Knowl.-Based Syst.* **2021**, *228*, 107293. [[CrossRef](#)]
30. Bodong, S.; Wiseong, J.; Chengmeng, L.; Khakichi, A. Economic management and planning based on a probabilistic model in a multi-energy market in the presence of renewable energy sources with a demand-side management program. *Energy* **2023**, *269*, 126549. [[CrossRef](#)]
31. Wynn, S.L.L.; Boonraksa, T.; Boonraksa, P.; Pinthurat, W.; Marungsri, B. Decentralized Energy Management System in Microgrid Considering Uncertainty and Demand Response. *Electronics* **2023**, *12*, 237. [[CrossRef](#)]
32. Sansa, I.; Boussaada, Z.; Bellaaj, N.M. Solar Radiation Prediction Using a Novel Hybrid Model of ARMA and NARX. *Energies* **2021**, *14*, 6920. [[CrossRef](#)]
33. Brahma, B.; Wadhvani, R. Solar irradiance forecasting based on deep learning methodologies and multi-site data. *Symmetry* **2020**, *12*, 1830. [[CrossRef](#)]
34. Jeon, B.k.; Kim, E.J. Next-day prediction of hourly solar irradiance using local weather forecasts and LSTM trained with non-local data. *Energies* **2020**, *13*, 5258. [[CrossRef](#)]
35. Husein, M.; Chung, I.Y. Day-ahead solar irradiance forecasting for microgrids using a long short-term memory recurrent neural network: A deep learning approach. *Energies* **2019**, *12*, 1856. [[CrossRef](#)]
36. Zafar, R.; Vu, B.H.; Husein, M.; Chung, I.Y. Day-Ahead Solar Irradiance Forecasting Using Hybrid Recurrent Neural Network with Weather Classification for Power System Scheduling. *Appl. Sci.* **2021**, *11*, 6738. [[CrossRef](#)]
37. Qing, X.; Niu, Y. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. *Energy* **2018**, *148*, 461–468. [[CrossRef](#)]
38. Srivastava, S.; Lessmann, S. A comparative study of LSTM neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Sol. Energy* **2018**, *162*, 232–247. [[CrossRef](#)]
39. Wang, F.; Yu, Y.; Zhang, Z.; Li, J.; Zhen, Z.; Li, K. Wavelet decomposition and convolutional LSTM networks based improved deep learning model for solar irradiance forecasting. *Appl. Sci.* **2018**, *8*, 1286. [[CrossRef](#)]
40. Ghimire, S.; Deo, R.C.; Raj, N.; Mi, J. Deep solar radiation forecasting with convolutional neural network and long short-term memory network algorithms. *Appl. Energy* **2019**, *253*, 113541. [[CrossRef](#)]
41. Huang, J.; Korolkiewicz, M.; Agrawal, M.; Boland, J. Forecasting solar radiation on an hourly time scale using a Coupled AutoRegressive and Dynamical System (CARDS) model. *Sol. Energy* **2013**, *87*, 136–149. [[CrossRef](#)]
42. Jiang, Y.; Long, H.; Zhang, Z.; Song, Z. Day-ahead prediction of bihourly solar radiance with a Markov switch approach. *IEEE Trans. Sustain. Energy* **2017**, *8*, 1536–1547. [[CrossRef](#)]
43. Ekici, B.B. A least squares support vector machine model for prediction of the next day solar insolation for effective use of PV systems. *Measurement* **2014**, *50*, 255–262. [[CrossRef](#)]
44. Yu, Y.; Cao, J.; Zhu, J. An LSTM short-term solar irradiance forecasting under complicated weather conditions. *IEEE Access* **2019**, *7*, 145651–145666. [[CrossRef](#)]
45. Wojtkiewicz, J.; Hosseini, M.; Gottumukkala, R.; Chambers, T.L. Hour-ahead solar irradiance forecasting using multivariate gated recurrent units. *Energies* **2019**, *12*, 4055. [[CrossRef](#)]
46. Zang, H.; Liu, L.; Sun, L.; Cheng, L.; Wei, Z.; Sun, G. Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations. *Renew. Energy* **2020**, *160*, 26–41. [[CrossRef](#)]
47. Gao, B.; Huang, X.; Shi, J.; Tai, Y.; Zhang, J. Hourly forecasting of solar irradiance based on CEEMDAN and multi-strategy CNN-LSTM neural networks. *Renew. Energy* **2020**, *162*, 1665–1683. [[CrossRef](#)]
48. Connor, J.; Atlas, L. Recurrent neural networks and time series prediction. In Proceedings of the IJCNN-91-Seattle international joint conference on neural networks, Seattle, WA, USA, 8–12 July 1991; Volume 1, pp. 301–306.
49. Brownlee, J. Time series prediction with lstm recurrent neural networks in python with keras. *Mach. Learn. Mastery* **2016**, *18*.
50. Moreno, J.J.M. Artificial neural networks applied to forecasting time series. *Psicothema* **2011**, *23*, 322–329.
51. Shen, Z.; Zhang, Y.; Lu, J.; Xu, J.; Xiao, G. A novel time series forecasting model with deep learning. *Neurocomputing* **2020**, *396*, 302–313. [[CrossRef](#)]

52. Bianchi, F.M.; Maiorino, E.; Kampffmeyer, M.C.; Rizzi, A.; Jenssen, R. *Recurrent Neural Networks for Short-Term Load Forecasting: An Overview and Comparative Analysis*; Springer: Berlin/Heidelberg, Germany, 2017.
53. Kumar, D.; Mathur, H.; Bhanot, S.; Bansal, R.C. Forecasting of solar and wind power using LSTM RNN for load frequency control in isolated microgrid. *Int. J. Model. Simul.* **2021**, *41*, 311–323. [[CrossRef](#)]
54. Li, D.; Tan, Y.; Zhang, Y.; Miao, S.; He, S. Probabilistic forecasting method for mid-term hourly load time series based on an improved temporal fusion transformer model. *Int. J. Electr. Power Energy Syst.* **2023**, *146*, 108743. [[CrossRef](#)]
55. DiPietro, R.; Hager, G.D. Deep learning: RNNs and LSTM. In *Handbook of Medical Image Computing and Computer Assisted Intervention*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 503–519.
56. Gupta, A.; Gurralla, G.; Sastry, P.S. Instability Prediction in Power Systems using Recurrent Neural Networks. In Proceedings of the IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 1795–1801.
57. Huo, Y.; Chen, Z.; Bu, J.; Yin, M. Learning assisted column generation for model predictive control based energy management in microgrids. *Energy Rep.* **2023**, *9*, 88–97. [[CrossRef](#)]
58. Cabrera-Tobar, A.; Massi Pavan, A.; Petrone, G.; Spagnuolo, G. A Review of the Optimization and Control Techniques in the Presence of Uncertainties for the Energy Management of Microgrids. *Energies* **2022**, *15*, 9114. [[CrossRef](#)]
59. Zhou, Y. Advances of machine learning in multi-energy district communities—mechanisms, applications and perspectives. *Energy AI* **2022**, *10*, 100187. [[CrossRef](#)]
60. Li, B.; Roche, R. Optimal scheduling of multiple multi-energy supply microgrids considering future prediction impacts based on model predictive control. *Energy* **2020**, *197*, 117180. [[CrossRef](#)]
61. Nyong-Bassey, B.E.; Giaouris, D.; Patsios, C.; Papadopoulou, S.; Papadopoulos, A.I.; Walker, S.; Voutetakis, S.; Seferlis, P.; Gadoue, S. Reinforcement learning based adaptive power pinch analysis for energy management of stand-alone hybrid energy storage systems considering uncertainty. *Energy* **2020**, *193*, 116622. [[CrossRef](#)]
62. Tang, W.; Zhang, Y.J. A model predictive control approach for low-complexity electric vehicle charging scheduling: Optimality and scalability. *IEEE Trans. Power Syst.* **2016**, *32*, 1050–1063. [[CrossRef](#)]
63. Karamanakos, P.; Geyer, T.; Kennel, R. A computationally efficient model predictive control strategy for linear systems with integer inputs. *IEEE Trans. Control Syst. Technol.* **2015**, *24*, 1463–1471. [[CrossRef](#)]
64. Zhang, L.; Zhuang, S.; Braatz, R.D. Switched model predictive control of switched linear systems: Feasibility, stability and robustness. *Automatica* **2016**, *67*, 8–21. [[CrossRef](#)]
65. Ayumi, V.; Rere, L.R.; Fanany, M.I.; Arymurthy, A.M. Optimization of convolutional neural network using microcanonical annealing algorithm. In Proceedings of the 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Malang, Indonesia, 15–16 October 2016; pp. 506–511.
66. Liu, C.T.; Wu, Y.H.; Lin, Y.S.; Chien, S.Y. Computation-performance optimization of convolutional neural networks with redundant kernel removal. In Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 27–30 May 2018; pp. 1–5.
67. Huang, R.; Wei, C.; Wang, B.; Yang, J.; Xu, X.; Wu, S.; Huang, S. Well performance prediction based on Long Short-Term Memory (LSTM) neural network. *J. Pet. Sci. Eng.* **2022**, *208*, 109686. [[CrossRef](#)]
68. Zhang, Y.; Hao, X.; Liu, Y. Simplifying long short-term memory for fast training and time series prediction. *J. Phys. Conf. Ser.* **2019**, *1213*, 042039. [[CrossRef](#)]
69. Schmidt, R.M. Recurrent neural networks (rnns): A gentle introduction and overview. *arXiv* **2019**, arXiv:1912.05911.
70. Amidi, A.; Amidi, S. *Vip Cheatsheet: Recurrent Neural Networks*; Stanford University: Stanford, CA, USA, 2018.
71. Ruder, S. An overview of gradient descent optimization algorithms. *arXiv* **2016**, arXiv:1609.04747.
72. Görges, D. Relations between model predictive control and reinforcement learning. *IFAC-PapersOnLine* **2017**, *50*, 4920–4928. [[CrossRef](#)]
73. Ying, X. An overview of overfitting and its solutions. *J. Phys. Conf. Ser.* **2019**, *1168*, 022022. [[CrossRef](#)]
74. Giaouris, D.; Papadopoulos, A.I.; Patsios, C.; Walker, S.; Ziogou, C.; Taylor, P.; Voutetakis, S.; Papadopoulou, S.; Seferlis, P. A systems approach for management of microgrids considering multiple energy carriers, stochastic loads, forecasting and demand side response. *Appl. Energy* **2018**, *226*, 546–559. [[CrossRef](#)]
75. Cavus, M.; Allahham, A.; Adhikari, K.; Giaouris, D. A Hybrid Method Based on Logic Control and Model Predictive Control for Synthesizing Controller for Flexible Hybrid Microgrid with Plug-and-Play Capabilities. Available online: <https://ssrn.com/abstract=4473008> (accessed on 15 July 2023).
76. Chicco, D.; Warrens, M.J.; Jurman, G. The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e623. [[CrossRef](#)]
77. Kong, X.; Du, X.; Xu, Z.; Xue, G. Predicting solar radiation for space heating with thermal storage system based on temporal convolutional network-attention model. *Appl. Therm. Eng.* **2023**, *219*, 119574. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.