Delft University of Technology

Delft Center for Systems and Control

Technical report 11-044

Optimal coordination of a multiple HVDC link system using centralized and distributed control*

P. Mc Namara, R.R. Negenborn, B. De Schutter, and G. Lightbody

If you want to cite this report, please use the following reference instead:

P. Mc Namara, R.R. Negenborn, B. De Schutter, and G. Lightbody, "Optimal coordination of a multiple HVDC link system using centralized and distributed control," *IEEE Transactions on Control Systems Technology*, vol. 21, no. 2, pp. 302–314, Mar. 2013.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: `https://www.dcsc.tudelft.nl`

*This report can also be downloaded via `https://pub.deschutter.info/abs/11_044.html`

Optimal coordination of a multiple HVDC link system using centralised and distributed control

Paul Mc Namara, *Student Member, IEEE*, Rudy R. Negenborn, Bart De Schutter, *Senior Member, IEEE*, Gordon Lightbody

***Abstract*—This paper presents both off-line and on-line optimisation techniques for the control of a multiple High Voltage Direct Current link power system. A frequency control scheme based on classical PID controllers is proposed and optimally tuned off-line using Particle Swarm Optimisation. The performance of this scheme is compared with the performance of a centralised Model Predictive Control (MPC) scheme, and a distributed MPC scheme that uses only local communications. The results illustrate that a significant performance improvement can be achieved using distributed MPC instead of classical control, illustrating the potential of distributed MPC for use in future power networks.**

***Index Terms*—Distributed control, Model Predictive Control, HVDC, multi-agent systems, Particle Swarm Optimisation.**

I. Introduction

POWER networks are large, complex, highly interconnected systems. As increasing demands are imposed on power networks, more advanced control techniques are needed in order to maintain network stability. This can be achieved by installing power system devices such as Flexible Alternating Current Transmission Systems (FACTSs) and High Voltage Direct Current (HVDC) links [1] while using more advanced control systems to maximise network efficiency. HVDC links allow for the efficient transmission of large quantities of power over long distances and can be used to improve transient stability and power system damping [1].

In [2], a multiple HVDC link system based on part of the Nordic power grid was presented. This is a nonlinear, MIMO dynamic system. The power generated in the system is kept constant and the modulation of the HVDC links alone is then used to restabilise the system after line faults. Thus there is a high level of interconnectivity between subsystems in this model. So far, centralised control techniques that were not based on optimisation, were used to control this system. These techniques focused on damping out oscillations between the AC connected areas, and were based on proportional control [2] and feedback linearisation [3]. However, while these controllers stabilised the system, there was significant scope for improving performance. Also, centralised control approaches for such systems may not always be possible, in the case of a deregulated power market for example.

Off-line optimisation techniques are often used to optimise parameters of a control system in order to maximise system performance based on certain criteria. Stochastic search techniques such as Simulated Annealing [4], Genetic Algorithms [5], and Particle Swarm Optimisation (PSO) [6] have proven to be efficient ways of finding globally optimal solutions for controller gains. Due to their ability to find global optima, Genetic Algorithms have proven to be particularly popular as a tool for finding optimal gains [7, 8, 5]. However, recently the advantages of using PSO over Genetic Algorithms for the optimisation of PID gains have been demonstrated, in terms of both the quality of and the efficiency with which a final solution can be found [6].

On-line optimisation techniques, such as Model Predictive Control (MPC) [9], use real-time optimisation in order to determine the control inputs for systems. One of the main advantages of MPC, over non-optimisation based techniques, is the systematic and intuitive manner in which constraints are incorporated into the control system and the fact that delays are naturally catered for. It is a mature technology at this stage, with stability and robustness analysis well established [10, 11, 12].

However, for large systems such as the electricity grid, it is often impractical to implement MPC from a central controller, due to computational constraints. Likewise it is often desirable or necessary to use a number of separate controllers, called agents, to control subsystems, e.g., in a deregulated power market several controllers may be responsible for the control of different sections of the power grid, or when power systems span several countries, countries will typically have separate controllers for their own sections of the grid.

There has been much research interest in recent years in distributed MPC [13, 14, 15], in which several agents communicate and cooperate with each other to approximate the behaviour of a centralised MPC agent. Lyapunov-based MPC techniques [15, 16], game-theoretic approaches [14], and iterative techniques based on the decomposition of the original control problem into several smaller problems [17, 18] have all been used to distribute the control amongst agents.

The performance of a centralised MPC scheme, which reaches a Pareto equilibrium, can be achieved in a distributed way if all agents in a system have all the information of a centralised MPC made available to them (assuming convex cost functions), i.e., the goals of all agents in a system, knowledge of the full state-space of the system, all system

Paul Mc Namara (paulmcn@rennes.ucc.ie) and Gordon Lightbody (g.lightbody@ucc.ie) are with the Control and Intelligent Systems Group in the Electrical and Electronic Engineering Department, University College Cork, Ireland.

Rudy R. Negenborn (r.r.negenborn@tudelft.nl) is with the Department of Marine and Transport Technology, Delft University of Technology, The Netherlands.

Bart De Schutter (b.deschutter@tudelft.nl) is with the Delft Center for Systems and Control, Delft University of Technology, The Netherlands.

constraints, etc. [18, 14]. However, in vast complex systems, such as electricity grids, the communication of this volume of information is impractical or may not even be possible, e.g., in a deregulated power market control agents may not be willing to share this level of information with other control agents.

However, when communication of interconnecting variables is allowed between adjacent agents, a Nash equilibrium can be achieved [14]. While Nash equilibrium seeking control systems, in general, do not achieve the performance of a Pareto equilibrium seeking control system, typically their performance will be significantly better than that achieved by a decentralised control system with no communications [18]. Indeed, promising results have already been achieved controlling power networks, which include FACTS devices, with control agents that only communicate with agents to which they are connected by a common variable [19, 18, 20].

In this paper, three optimisation-based control techniques are proposed for the control of the Nordic multiple HVDC link system. The controllers used are an off-line PSO-optimised PID controller, a centralised MPC controller, and a distributed MPC controller that uses only local communications [17].

Typically distributed MPC systems will try to reach consensus on interconnecting variables between agents. However in the example in this paper all 4 agents share the same 2 control inputs. Thus, we elaborate on how the control system in [17] is used in order to coordinate the agents' responses for control inputs that are common to multiple agents.

As distributed MPC provides a scalable MPC architecture that could realistically be used in large-scale power systems, it is interesting to see how its performance compares with that of an optimised centralised classical PID controller, given the widespread use of PID controllers within the power industry. As part of this analysis, however, it is necessary to take into account the computational and communication overhead associated with distributed MPC to get a better idea of the trade-offs involved when using distributed MPC.

This paper is organised as follows. The multiple HVDC link system is presented in Section II. The off-line PSO optimisation of the PID gains of the multiple HVDC link controller is presented in Section III. MPC and distributed MPC are introduced in Section IV. Simulation studies compare the performance of the different approaches in Section V. Section VI contains conclusions and directions for future research.

II. The multiple HVDC link system

The continuous-time dynamics of the multiple HVDC link system under study here are described in this section.

A. Multiple HVDC link system description

The system, which is based on the multiple HVDC link system between Denmark, Norway, and Sweden, is depicted in Fig. 1 [21]. It consists of 4 buses with their own generation and loads. Both Alternating Current (AC) and HVDC links connect the buses. The HVDC links are of the Line Commutated Converter type [22]. Generation capacities and loads are kept constant in this paper. Large amounts of power are transferred from bus 2, which has the largest generation capacity, to bus 4, which has the largest power load.

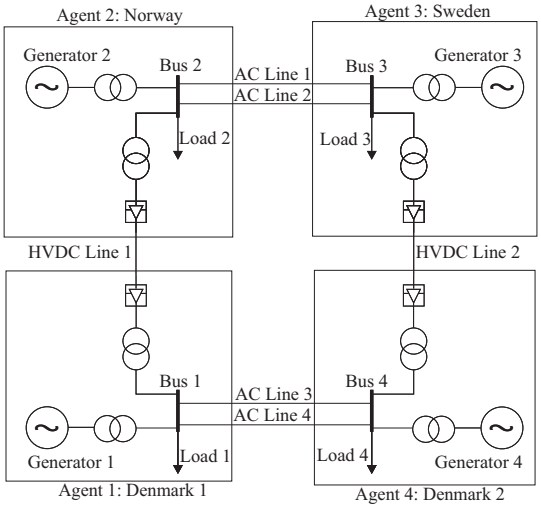


Fig. 1. The multiple HVDC link system with areas controlled by agents [21].

B. Modelling

The classical swing equations for a generator a are [1]:

$$\frac{\mathrm{d}}{\mathrm{d}t}\delta_{\mathrm{r}a}(t) = \omega_0 \Delta\omega_{\mathrm{r}a}(t) \tag{1}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\omega_{\mathrm{r}a}(t) = \frac{1}{2H_a}(P_{\mathrm{m}a}(t) - P_{\mathrm{G}a}(t) - D_a\Delta\omega_{\mathrm{r}a}(t)), \tag{2}$$

where $\delta_{\mathrm{r}a}(t)$ is the rotor angle (rad), H_a is the inertial constant (s), $\omega_{\mathrm{r}a}(t)$ is the rotor speed (per unit), $\Delta\omega_{\mathrm{r}a}(t) = \omega_{\mathrm{r}a}(t) - 1$ is the rotor speed deviation (per unit), ω_0 is the base rotor speed (rad/s), $P_{\mathrm{m}a}(t)$ and $P_{\mathrm{G}a}(t)$ are the mechanical and generated power (per unit), respectively, and D_a is the damping factor (per unit).

The current injected by generator a, $\overrightarrow{I}_{\mathrm{g}a}(t)$, is given by:

$$\overrightarrow{I}_{\mathrm{g}a}(t) = \frac{\overrightarrow{E}'_{\mathrm{q}a}(t) - \overrightarrow{U}_a(t)}{jx'_{\mathrm{d}a}}, \tag{3}$$

where $\overrightarrow{E}'_{\mathrm{q}a}(t) = E'_{\mathrm{q}a}(t)\angle\delta_a(t)$ is the internal voltage (per unit) with magnitude $E'_{\mathrm{q}a}(t)$ and angle $\delta_a(t)$, $\overrightarrow{U}_a(t) = U_a(t)\angle\theta_a(t)$ is the voltage (per unit) at the bus to which the generator is connected with magnitude $U_a(t)$ and angle $\theta_a(t)$, and $x'_{\mathrm{d}a}$ is the d-axis transient reactance. All variables are defined as in the standard reference work [1]. The generated power is then given by:

$$P_{\mathrm{G}a}(t) = \mathbb{R}[\overrightarrow{E}_{\mathrm{g}a}(t)\,\overrightarrow{I}^{*}_{\mathrm{g}a}(t)]. \tag{4}$$

Equations (1) and (2) of the classical model of a synchronous generator assume that $E'_{\mathrm{q}a}(t)$ is constant [1]. These classical equations are suitable for analysis of power oscillations and transient stability studies.

A π-model representation [1] of the AC links is used, as in Fig. 2, where X_{L_i} is the line reactance, X_{S_i} is the shunt reactance, X_{G_i} is a ground reactance (used here to simulate 3-phase to ground faults in the middle of line i), X_{B_i} is a

reactance (used here to simulate a line break), U_{L_1} and U_{L_2} are the voltages on either side of line i, and U_{m_i} is the voltage at the middle of the line connected to ground through X_{G_i}. The value of X_{G_i} decreases from ∞ in the non-fault state to 0 in the case of the 3-phase fault. As the faults happen in the middle of the line, the line reactance is divided in half on either side of the ground fault "line". The value of X_{B_i} increases from a value of 0 to ∞ when a line break is applied.

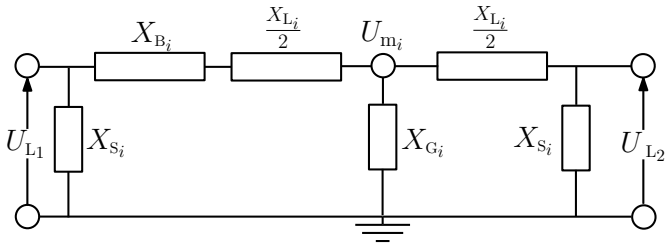


Fig. 2. π-model of lines.

A number of simplifications are made to the power system model, subsequently reducing the complexity of the MPC problem proposed in the next section, and making it faster, without neglecting the system dynamics most relevant to system stability.

- Loads are modelled as constant impedances.
- HVDC power transmission is considered instantaneous.
- The HVDC reactive power is taken as proportional to HVDC active power.

The details of these simplifications are given in the following paragraphs.

Loads are modelled as constant impedances in the impedance matrix, i.e. with $\overrightarrow{S}_a = P_a^{\mathrm{LL}} + jQ_a^{\mathrm{LL}}$, the consumed complex load power in VA, the load impedance $\overrightarrow{X}_a^{\mathrm{LL}}$ is given by

$$\overrightarrow{X}_a^{\mathrm{LL}} = \frac{\overrightarrow{U}_a \overrightarrow{U}_a^*}{\overrightarrow{S}_a^*}. \tag{5}$$

The HVDC link model in [21] is used to simplify the representation of the system dynamics. This idealised version of the HVDC link assumes instantaneous, lossless power delivery and power factors that are equal on both the inverter and rectifier sides. Thus, when active power injections from a HVDC link j are applied to a bus in this model, the bus receives $+P_j^{\mathrm{DC}}$, where P_j^{DC} is the active power in the HVDC link. Equally, the bus from which the HVDC active power is being sent then loses P_j^{DC}. This model is further simplified by assuming that a simple controller structure is used as in [22], such that $Q_j^{\mathrm{DC}} = q_{\mathrm{r}_j} P_j^{\mathrm{DC}}$, where q_{r_j} is a constant, and Q_j^{DC} is the reactive HVDC power in HVDC link j. An additional simplification adopted in this paper is to directly calculate and apply the HVDC powers. However, in a real system, currents are injected that are calculated from these powers [1].

The internal node representation is used to model the system dynamics [21] as it allows the power system to be represented using a system of first-order differential equations. This is done by rearranging the impedance matrix of the power network so as to find its generator currents in terms of the network voltages and HVDC currents, and then substituting the values for these currents into the generator swing equations. This is demonstrated in the following paragraphs. To do this it is assumed that P_{m_a} is constant and that the loads are modelled as constant impedances.

An impedance matrix gives the relationship between the voltage nodes and currents in a power system. Lines and loads are then represented by impedances in this matrix. Consider the a^{th} bus, with n voltage nodes and m HVDC links connected to it. Then using Kirchhoff's current law:

$$\sum_{l=1}^{n} \vec{Y}_{a,l} \vec{U}_l - \vec{I}_{\mathrm{g}a} - \sum_{j=1}^{m} \vec{I}_{\mathrm{DC}_{a,\,j}} = 0, \tag{6}$$

where $\vec{Y}_{a,l} = \frac{1}{\vec{X}_{a,l}}$, where $\vec{X}_{a,l}$ is the impedance between bus a and voltage node l and $\vec{I}_{\mathrm{DC}_{a,\,j}}$ is the current injected into bus a from HVDC link j. This is repeated for all voltage nodes in the network.

Using (6), an impedance matrix is constructed, based on models for the AC lines, the generators, and the loads, as described in the previous paragraphs:

$$\begin{pmatrix} \boldsymbol{I}_{\mathrm{g}} \\ \boldsymbol{I}_{\mathrm{DC}} \\ 0 \end{pmatrix} = \begin{pmatrix} \boldsymbol{Y}_{\mathrm{A}} & \boldsymbol{Y}_{\mathrm{B}} & 0 \\ \boldsymbol{Y}_{\mathrm{C}} & \boldsymbol{Y}_{\mathrm{D}} & \boldsymbol{Y}_{\mathrm{E}} \\ 0 & \boldsymbol{Y}_{\mathrm{F}} & \boldsymbol{Y}_{\mathrm{G}} \end{pmatrix} \begin{pmatrix} \boldsymbol{E} \\ \boldsymbol{U} \\ \boldsymbol{U}_{\mathrm{m}} \end{pmatrix}, \tag{7}$$

where $\boldsymbol{I}_{\mathrm{g}} = [\vec{I}_{\mathrm{g}_1}, \ldots, \vec{I}_{\mathrm{g}_n}]^{\mathrm{T}}$, $\boldsymbol{E} = [\vec{E}'_{\mathrm{q}_1}, \ldots, \vec{E}'_{\mathrm{q}_n}]^{\mathrm{T}}$, $\boldsymbol{U} = [\vec{U}_1, \ldots, \vec{U}_n]^{\mathrm{T}}$ and $\boldsymbol{U}_{\mathrm{m}} = [\vec{U}_{\mathrm{m}_1}, \ldots, \vec{U}_{\mathrm{m}_b}]^{\mathrm{T}}$ where b is the number of AC lines in the system and $\boldsymbol{I}_{\mathrm{DC}} = [\vec{I}_{\mathrm{DC}_{1,\,1}}, \ldots, \vec{I}_{\mathrm{DC}_{n,\,m}}]^{\mathrm{T}}$.

From (7) the following can be found for $\boldsymbol{I}_{\mathrm{g}}$ in terms of $\boldsymbol{I}_{\mathrm{DC}}$ and $\boldsymbol{E}$:

$$\begin{aligned} \boldsymbol{I}_{\mathrm{g}} &= (\boldsymbol{Y}_{\mathrm{A}} - \boldsymbol{Y}_{\mathrm{B}}(\boldsymbol{Y}_{\mathrm{D}} - \boldsymbol{Y}_{\mathrm{E}}\boldsymbol{Y}_{\mathrm{G}}^{-1}\boldsymbol{Y}_{\mathrm{F}}))\boldsymbol{E} \\ &\quad + \boldsymbol{Y}_{\mathrm{B}}(\boldsymbol{Y}_{\mathrm{D}} - \boldsymbol{Y}_{\mathrm{E}}\boldsymbol{Y}_{\mathrm{G}}^{-1}\boldsymbol{Y}_{\mathrm{F}})^{-1}\boldsymbol{I}_{\mathrm{DC}} \\ &= (\boldsymbol{G} + j\boldsymbol{B})\boldsymbol{E} + \boldsymbol{Y}_{\mathrm{DC}}\boldsymbol{I}_{\mathrm{DC}} \end{aligned} \tag{8}$$

where $\boldsymbol{G}$=$\mathbb{R}[\boldsymbol{Y}_{\mathrm{A}} - \boldsymbol{Y}_{\mathrm{B}}(\boldsymbol{Y}_{\mathrm{D}} - \boldsymbol{Y}_{\mathrm{E}}\boldsymbol{Y}_{\mathrm{G}}^{-1}\boldsymbol{Y}_{\mathrm{F}})]$, $\boldsymbol{B}$=$\mathbb{I}[\boldsymbol{Y}_{\mathrm{A}} - \boldsymbol{Y}_{\mathrm{B}}(\boldsymbol{Y}_{\mathrm{D}} - \boldsymbol{Y}_{\mathrm{E}}\boldsymbol{Y}_{\mathrm{G}}^{-1}\boldsymbol{Y}_{\mathrm{F}})]$, and $\boldsymbol{Y}_{\mathrm{DC}}$=$\boldsymbol{Y}_{\mathrm{D}} - \boldsymbol{Y}_{\mathrm{E}}\boldsymbol{Y}_{\mathrm{G}}^{-1}\boldsymbol{Y}_{\mathrm{F}}$.

The following swing equation for generator a can be derived using (2), (4), and (8):

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}\omega_{\mathrm{r}_a} = \frac{1}{2H_a}\Big(P_{\mathrm{m}_a} - G_{a,a}E_{\mathrm{q}_a}^{'2} - \\ &\sum_{\substack{l=1 \\ l\neq a}}^{n} E_{\mathrm{q}_a}^{'} E_{\mathrm{q}_l}^{'} (G_{a,l}\cos(\delta_{\mathrm{r}_a} - \delta_{\mathrm{r}_l}) + B_{a,l}\sin(\delta_{\mathrm{r}_a} - \delta_{\mathrm{r}_l})) \\ &\qquad + g_{a,1}P_1^{\mathrm{DC}} + \ldots + g_{a,m}P_m^{\mathrm{DC}} - D_a\Delta\omega_{\mathrm{r}_a}\Big), \end{aligned} \tag{9}$$

where $g_{a,j}$ is the coefficient of the contribution of the power injections from HVDC link j at bus a.

C. Definition of an agent

For clarity the definition of an agent, as understood in this paper, will now be provided. An agent is defined here as an entity responsible for the control of a system or subsystem, with access to the current state of the system or subsystem it controls. Agents have access to a model of the local system

or subsystem and in the distributed case, agents are able to communicate with other agents who share a common variable. Agents compute values for their control inputs at discrete time steps based on the information available to them.

D. Control goals for the multiple HVDC link system

It is desirable to install a control system that maintains the rotor frequencies as close as possible to 1 pu at all times for the multiple link HVDC system. In this work the base frequency is 50 Hz, the minimum allowable frequency is 49.2 Hz, and the maximum allowable frequency is 50.8 Hz [23]. In per unit terms it is therefore desired to keep the frequency between 0.984 pu and 1.016 pu. The total generated power equals the total consumed power in the network, and so the modulation of the HVDC link powers alone should be sufficient to restabilise the generator frequencies at each bus, and return them to their desired setpoints after frequency deviations are incurred due to line fault disturbances.

Also the need for a multi-agent approach for controlling this system could arise. Consider that 2 different agents are responsible for the areas in Denmark due to market deregulation; usually controllers do not cross borders and so another 2 agents could be responsible for the control of the areas in Norway and Sweden, leading to a situation where 4 agents are responsible for the control of this power network. Also, these agents all share the same inputs, and the AC-line connected agents are affected by their connected agent's generator rotor positions, and so the control problems of all agents in the system are highly interconnected. For this reason several different centralised and non-centralised control schemes are applied to the system in this paper.

III. Off-line PSO Optimisation of a centralised PID control scheme

Proportional controllers were initially proposed in [21] for control of the multiple HVDC link system, using the control scheme presented in Fig. 3. The proportional gains and filter bandwidths were chosen based on trial and error, observing which combinations of values gave a good damping response. However there is the potential for improved control performance by using Proportional, Integral and Derivative (PID) controllers, and then optimising both the controller gains and the filter bandwidths, based on simulation runs of the power network, using a suitable tuning criterion.

As simulations of the power network, which is represented by a non-linear model, are used to provide the input to the tuning criterion, the derivation of the relationship between the PID gains and the tuning criterion is non-trivial and so a derivative-free optimisation technique is used. Due to the non-convex nature of these tuning problems, it is desirable to use an optimisation technique that does not get trapped in local minima.

Stochastic search techniques such as Simulated Annealing [4], Genetic Algorithms [5], and Particle Swarm Optimisation [6] have previously been quite successful in finding optimal controller gains. PSO has been shown to outperform other stochastic search algorithms, such as Genetic Algorithms, in terms of both the quality of the solutions found and the time needed to find them [6]. Hence, PSO is used in this work.

A. Particle Swarm Optimisation

Particle Swarm Optimisation (PSO) is a stochastic optimisation technique based on the social behaviour of swarms of flocking animals [24]. It is suitable for the optimisation of convex, non-convex, continuous and discontinuous surfaces.

In PSO a population of P particles, each of dimension d, are initially distributed across the parameter space. The q^{th} particle at the i^{th} iteration of the PSO algorithm, has a position $\boldsymbol{x}_q(i)$, and associated cost $x_q^{\text{c}}(i)$. Each of these particles has a memory of its own previous best position $\boldsymbol{p}_q^{\text{b}}(i)$ and an associated cost $p_q^{\text{c,b}}(i)$. Here $p^{\text{g}}(i)$, the global best position, is the particle position associated with the best cost $p^{\text{c,g}}(i)$ that has been found previously across the population of particles. The q^{th} particle position is then updated, biased towards both the global best position and its previous best position. The PSO algorithm (in the case of the minimisation of a cost function) is as follows:

1) Initialise a population of P particles in d dimensions, within upper and lower bounds in each dimension, in the cost function space.
2) Evaluate the cost function values at each of the P particles' positions.
3) If for particle q, $x_q^{\text{c}}(i)<p_q^{\text{c,b}}(i-1)$, then let $\boldsymbol{p}_q^{\text{b}}(i)=\boldsymbol{x}_q(i)$.
   If $x_q^{\text{c}}(i)<p^{\text{c,g}}(i-1)$, let $\boldsymbol{p}^{\text{g}}(i)=\boldsymbol{x}_q(i)$.
   If $x_q^{\text{c}}(i)\geq p_q^{\text{c,b}}(i-1)\geq p^{\text{c,g}}(i-1)$, then $\boldsymbol{p}_q^{\text{b}}(i)$ and $\boldsymbol{p}^{\text{g}}(i)$ remain at the same positions as in iteration $i-1$.
4) The velocity, $\boldsymbol{v}_{\text{q}}(i)$, and position $\boldsymbol{x}_q(i)$, of particle q at the i^{th} iteration of the PSO algorithm are updated for the next iteration as follows:

$$\begin{aligned}\boldsymbol{v}_q(i+1) = \omega\boldsymbol{v}_q(i) &+ c_1\boldsymbol{r}_1(i)\circ\left(\boldsymbol{p}_q^{\text{b}}(i)-\boldsymbol{x}_q(i)\right)\\ &+ c_2\boldsymbol{r}_2(i)\circ\left(\boldsymbol{p}^{\text{g}}(i)-\boldsymbol{x}_q(i)\right)\end{aligned} \tag{10}$$

$$\boldsymbol{x}_q(i+1) = \boldsymbol{x}_q(i) + \boldsymbol{v}_{q_{\text{app}}}(i+1) \tag{11}$$

where $\circ$ denotes the Schur product, $\boldsymbol{r}_1(i)$ and $\boldsymbol{r}_2(i)$ are random vectors with entries uniformly distributed in the interval [0,1], the positive scalar ω is the inertial weight which controls the exploration and exploitation in the search space, c_1 and c_2 are acceleration constants called the cognition and social components, respectively, and $\boldsymbol{v}_{q_{\text{app}}}$ is the applied particle velocity.

Applied particle velocities are bounded by $\boldsymbol{v}_{q_{\min}}\leq\boldsymbol{v}_{q_{\text{app}}}\leq\boldsymbol{v}_{q_{\max}}$ where $\boldsymbol{v}_{q_{\min}}$ and $\boldsymbol{v}_{q_{\max}}$ are the lower and upper bounds on particle velocities, respectively. If updated velocities exceed the bounds, the applied velocity, $\boldsymbol{v}_{q_{\text{app}}}$, is taken at the upper or lower bound, i.e., if $\boldsymbol{v}_q<\boldsymbol{v}_{q_{\min}}$, let $\boldsymbol{v}_{q_{\text{app}}}=\boldsymbol{v}_{q_{\min}}$; if $\boldsymbol{v}_q>\boldsymbol{v}_{q_{\max}}$, let $\boldsymbol{v}_{q_{\text{app}}}=\boldsymbol{v}_{q_{\max}}$; else let $\boldsymbol{v}_{q_{\text{app}}}=\boldsymbol{v}_q$.
5) Repeat steps (2)-(4) until certain termination criteria are performed, e.g., a maximum amount of iterations are met, $\boldsymbol{p}^{\text{g}}(i)$ has not changed for a given number of iterations, etc.

In [25] it was shown that good convergence properties could be obtained for the PSO, using the following parameter

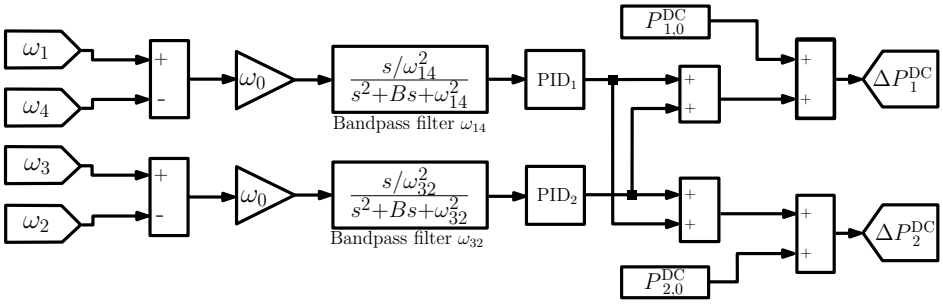


Fig. 3. PID control strategy used to control the multiple link HVDC system [21]

selection; $\omega = 0.6$, c_1=c_2=1.7. These parameter values were selected for this work so as to give a good convergence performance on the final set of values chosen by the PSO.

B. Control strategy and optimisation

The control strategy proposed in [21] for the control of the multiple HVDC link system can be seen in Fig. 3. The generator speed deviation between the AC connected generators is passed to a bandpass filter with the following transfer function:

$$G(s) = \frac{s/\omega_{\mathrm{c}}^2}{s^2 + Bs + \omega_{\mathrm{c}}^2} \tag{12}$$

where B is the -3dB bandwidth in rad/s and ω_{c} is the centre frequency of the filter in rad/s. In the diagram $P_{1,0}^{\mathrm{DC}}$ and $P_{2,0}^{\mathrm{DC}}$ are the initial operating points for P_1^{DC} and P_2^{DC}, respectively.

The bandpass filters are centred on the major oscillatory modes between buses 1 and 4 (ω_{14} rad/s) and buses 2 and 3 (ω_{32} rad/s). PID controllers are then used to modulate the HVDC links in order to dampen the oscillations in the AC line speed deviations at the major oscillatory modes.

PSO is used to optimise the PID gains and filter bandwidths. A PSO particle is made to correspond to a vector of the controller parameters. Each particle runs a simulation of the multiple link HVDC system with the PID controller, using its current position to determine the controller gains and bandwidths. The particle cost is obtained by recording the generator frequencies over the course of the simulation and passing these to an evaluation criterion. The PSO is then allowed to run as described previously.

The criterion used here to evaluate the cost for a simulation run is the integral of the square of the error (ISE) [6] given by:

$$\mathrm{ISE} = \sum_{a=1}^{n} \sum_{k=1}^{k_{\mathrm{f}}} (\omega_{\mathrm{r}_a}(k) - 1)^2, \tag{13}$$

where there are n agents, and k_{f} is the number of samples in one system simulation run. The ISE is used to reward behaviours that minimise errors over the simulation, with penalties increasing quadratically as the errors increase in magnitude. This criterion is used as it is desirable to minimise the maximum size of frequency deviations in power systems.

IV. Model Predictive Control

Model Predictive Control [9] is an advanced control technique based on real-time optimisation. It uses predictions, based on a suitable model, in order to provide optimal control inputs to a system. One of its main advantages over other control techniques is its ability to incorporate constraints into its formulation. It is at this stage a mature technology with feasibility, stability and robustness proofs well established [10].

A. Description and state space prediction

In MPC a control agent uses a discrete-time system model that predicts the system's future trajectory over a prediction horizon in order to calculate the optimal inputs over this horizon. Only the input for the first time step is applied. At the next time step a new action is determined. MPC is often called Receding Horizon Control due to the prediction horizon moving forward each time step.

A system consisting of n subsystems is considered, where each subsystem consists of a set of nodes and the interconnections between these nodes. Subsystems are assumed to be non-overlapping, i.e., nodes do not appear in 2 different subsystems. A discrete-time, linear, time-invariant state-space model is used to model the subsystem dynamics. This is given as follows:

$$\boldsymbol{x}_a(k+1) = \boldsymbol{A}_a\boldsymbol{x}_a(k) + \boldsymbol{B}_a\boldsymbol{u}_a(k) + \boldsymbol{D}_a\boldsymbol{d}_a(k) + \boldsymbol{V}_a\boldsymbol{v}_a(k) \tag{14}$$

$$\boldsymbol{y}_a(k) = \boldsymbol{C}_a^{\mathrm{x}}\boldsymbol{x}_a(k) + \boldsymbol{C}_a^{\mathrm{u}}\boldsymbol{u}_a(k) + \boldsymbol{C}_a^{\mathrm{d}}\boldsymbol{d}_a(k) + \boldsymbol{C}_a^{\mathrm{v}}\boldsymbol{v}_a(k), \tag{15}$$

where $\boldsymbol{x}_a(k)$ is the state of subsystem a, $\boldsymbol{u}_a(k)$ are local subsystem inputs, $\boldsymbol{d}_a(k)$ are known disturbances, $\boldsymbol{y}_a(k)$ are subsystem outputs, and $\boldsymbol{v}_a(k)$ are external inputs from other subsystems that influence subsystem i at sample time k.

To simplify notation, the prediction vector, over a horizon N is first introduced. For a general vector $\boldsymbol{z}$, its prediction vector is $\tilde{\boldsymbol{z}}(k) = [\boldsymbol{z}^{\mathrm{T}}(k) \dots \boldsymbol{z}^{\mathrm{T}}(k+N-1)]^{\mathrm{T}}$. State predictions for subsystem i over the prediction horizon are then determined using (14) as follows:

$$\tilde{\boldsymbol{x}}_a(k+1) = \boldsymbol{A}_a^{\mathrm{f}}\boldsymbol{x}_a(k) + \boldsymbol{B}_a^{\mathrm{f}}\tilde{\boldsymbol{u}}_a(k) + \boldsymbol{D}_a^{\mathrm{f}}\tilde{\boldsymbol{d}}_a(k) + \boldsymbol{V}_a^{\mathrm{f}}\tilde{\boldsymbol{v}}_a(k) \tag{16}$$

where $\boldsymbol{A}_a^{\mathrm{f}}$, $\boldsymbol{B}_a^{\mathrm{f}}$, $\boldsymbol{D}_a^{\mathrm{f}}$, $\boldsymbol{V}_a^{\mathrm{f}}$ are the state space prediction matrices. The derivation of these matrices is well established in the literature [9].

B. MPC formulations

In a system of n subsystems, with agents $1, \dots, n$, assume a situation where agent a operates individually without communication with other agents. Suppose for now that $\boldsymbol{x}_a(k)$,

$\tilde{\boldsymbol{d}}_a(k)$, and $\tilde{\boldsymbol{v}}_a(k)$ are all available. The following optimisation problem is then solved at each time step:

$$\begin{aligned}\tilde{\boldsymbol{u}}_a(k) = \arg\min_{\tilde{\boldsymbol{u}}_a(k)} J_a^{\text{local}}(\boldsymbol{x}_a(k), \tilde{\boldsymbol{u}}_a(k), \tilde{\boldsymbol{d}}_a(k), \tilde{\boldsymbol{v}}_a(k)) \\ \text{subject to } \tilde{\boldsymbol{u}}_a(k) \in \Omega_a,\ \tilde{\boldsymbol{x}}_a(k) \in \theta_a,\end{aligned} \tag{17}$$

where Ω_a and θ_a are the sets of admissible inputs and states respectively, for subsystem a. The local cost of subsystem a at the k^{th} sample time is (henceforth, $J_a^{\text{local}}(\boldsymbol{x}_a(k), \tilde{\boldsymbol{u}}_a(k), \tilde{\boldsymbol{d}}_a(k), \tilde{\boldsymbol{v}}_a(k))$ is denoted as $J_a^{\text{local}}(k)$),

$$J_a^{\text{local}}(k) = \sum_{p=0}^{N-1} J_a^{\text{stage}}(k,p). \tag{18}$$

Here $J_a^{\text{stage}}(k,p)$ is the cost at the p^{th} step of the prediction horizon for subsystem i at sample step k. This is generally set up as a weighted sum of the square of the errors at the p^{th} prediction step.

However, when many subsystems are interconnected, then knowledge of $\tilde{\boldsymbol{v}}_a(k)$ cannot be assumed, as $\tilde{\boldsymbol{v}}_a(k)$ is dependent on the dynamics of other subsystems. Hence, subsystems must reach a consensus on values for interconnecting variables.

Let there be a set of m_a agents, with indices $j \in \mathcal{N}_a$, that are connected to agent a. The interconnecting input vector, $\boldsymbol{w}_{ja}^{\text{in}}$, is defined as the vector of inputs to control problem a from agent j and the interconnecting output vector $\boldsymbol{w}_{ja}^{\text{out}}$ is defined as the vector of outputs to control problem j from agent a.

Centralised, decentralised and distributed control schemes based on the above formulation will now be presented.

1) Centralised MPC: In centralised MPC, instead of each subsystem having its own control agent, one central agent controls the whole system, solving all the individual subsystem MPC problems simultaneously. For a system of n subsystems, the combined overall optimisation problem can be formed as follows:

$$\begin{aligned}\min_{\tilde{\boldsymbol{u}}_1(k),\ldots,\tilde{\boldsymbol{u}}_n(k)} \sum_{i=1}^{n} J_a^{\text{local}}(k) \\ \text{subject to } \tilde{\boldsymbol{u}}_a(k) \in \Omega_a,\ \tilde{\boldsymbol{x}}_a(k) \in \theta_a,\end{aligned} \tag{19}$$

for $a = 1, \ldots, n$,

and subject to the following equality constraints,

$$\tilde{\boldsymbol{w}}_{ja}^{\text{in}}(k) = \tilde{\boldsymbol{w}}_{aj}^{\text{out}}(k),\ \text{for } j \in \mathcal{N}_a, \tag{20}$$

i.e., all interconnecting variables are made equal to each other over the prediction horizon according to the dynamics of each subsystem, as given in (16).

However, often the implementation of centralised MPC can be impractical due to technical constraints, e.g., the computational load being too large or several separate agents may be responsible for the control of different connected subsystems, such as when different controllers in different countries control sections of connected power grids. Therefore several agents are used to control different subsystems and the behaviour of these agents together should approximate the behaviour of the centralised MPC.

2) Decentralised MPC: Decentralised MPC schemes assume that interconnected subsystems interact weakly and so ignore the effects of interactions with other subsystems in their MPC problems. Agents do not communicate with each other and independently solve an optimisation problem similar to (17) for each subsystem, without seeking to achieve consensus amongst connected subsystems. However, ignoring these interactions between subsystems can lead to highly suboptimal behaviour [18].

3) Distributed case: In distributed MPC systems, agents communicate with each other in order to coordinate their control actions. An augmented Lagrangian formulation can be formulated from (19) to incorporate the equality constraints (20) into the cost function. In [17] the quadratic terms of the augmented Lagrangian formulation are distributed across the agents using Block Coordinate Descent. In this approach, one agent at a time optimises values for its inputs, $\tilde{\boldsymbol{u}}_a(k)$, and its desired interconnecting input variables $\tilde{\boldsymbol{w}}_{ja}^{\text{in}}(k)$, for each $j \in \mathcal{N}_a$. The optimisation problem of agent a, for the l^{th} iteration of the distributed MPC cycle, at the k^{th} time step is:

$$\min_{\tilde{\boldsymbol{u}}_a(k), \{\tilde{\boldsymbol{w}}_{ji}^{\text{in}}(k): j \in \mathcal{N}_a\}} \left(J_a^{\text{local}}(k) + \sum_{j \in \mathcal{N}_a} J_a^{\text{inter}}(k,l) \right) \tag{21}$$

where $J_a^{\text{inter}}(k,l)$ is the cost associated with the inter-agent coordination given by:

$$\begin{aligned}J_a^{\text{inter}}(k,l) = &\begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{ja}^{\text{in}}(l) \\ -\tilde{\boldsymbol{\lambda}}_{aj}^{\text{in}}(l) \end{bmatrix}^{\text{T}} \begin{bmatrix} \tilde{\boldsymbol{w}}_{ja}^{\text{in}}(k) \\ \tilde{\boldsymbol{w}}_{ja}^{\text{out}}(k) \end{bmatrix} \\ &+ \frac{c}{2} \left\| \begin{bmatrix} \tilde{\boldsymbol{w}}_{aj,\text{prev}}^{\text{in}}(l) - \tilde{\boldsymbol{w}}_{ja}^{\text{out}}(k) \\ \tilde{\boldsymbol{w}}_{aj,\text{prev}}^{\text{out}}(l) - \tilde{\boldsymbol{w}}_{ja}^{\text{in}}(k) \end{bmatrix} \right\|_2^2,\end{aligned} \tag{22}$$

where c is a positive constant and $\tilde{\boldsymbol{\lambda}}_{ja}^{\text{in}}(l)$ is the Lagrange multiplier associated with the interconnecting constraint $\tilde{\boldsymbol{w}}_{ja}^{\text{in}}(k) = \tilde{\boldsymbol{w}}_{aj}^{\text{out}}(k)$ at iteration l.

Each agent optimises this cost in a serial fashion, communicating the interconnecting variables with its neighbours. The values $\tilde{\boldsymbol{w}}_{aj,\text{prev}}^{\text{out}}(l)$, $\tilde{\boldsymbol{w}}_{aj,\text{prev}}^{\text{in}}(l)$ are taken as the most recently updated values of $\tilde{\boldsymbol{w}}_{aj}^{\text{out}}(k)$ and $\tilde{\boldsymbol{w}}_{aj}^{\text{in}}(k)$ respectively.

One optimisation cycle is completed when all agents have performed an optimisation. When the optimisation cycle is finished, the Lagrange multipliers are updated as follows:

$$\tilde{\boldsymbol{\lambda}}_{ja}^{\text{in}}(l+1) = \tilde{\boldsymbol{\lambda}}_{ja}^{\text{in}}(l) + c\left(\tilde{\boldsymbol{w}}_{ja}^{\text{in}}(k) - \tilde{\boldsymbol{w}}_{aj}^{\text{out}}(k)\right), \tag{23}$$

Iterations are continued until:

$$\begin{aligned}&||\tilde{\boldsymbol{\lambda}}_{ja}^{\text{in}}(l+1) - \tilde{\boldsymbol{\lambda}}_{ja}^{\text{in}}(l)||_\infty \leq \epsilon \\ &\text{for } a = 1, \ldots, n \text{ and } j \in \mathcal{N}_a\end{aligned} \tag{24}$$

where ϵ is a specified tolerance and $\|.\|_\infty$ denotes the infinity norm.

C. Application to shared inputs

In typical control applications, agents have their own local control inputs which are not shared between agents. However, in the application in this paper, all 4 agents must determine

actions for the 2 control inputs, $P_1^{\rm DC}(k)$ and $P_2^{\rm DC}(k)$. In other circumstances different agents' local inputs may be coupled, for example, via the objective function or through the system dynamics.

The algorithm in [17], used in this paper, naturally extends to such cases in the following way. Agents create a duplicate variable vector, $\tilde{\boldsymbol{w}}_a^{\rm u}(k)$, for agent a, of the control inputs, $\tilde{\boldsymbol{u}}(k)$, and then try to form consensus on these duplicate variables. These duplicate variables are then treated as local control inputs by each of the agents. Equality constraints are then placed on the duplicate variables as follows $\tilde{\boldsymbol{w}}_1^{\rm u}(k)$=$\tilde{\boldsymbol{w}}_2^{\rm u}(k)$, $\tilde{\boldsymbol{w}}_2^{\rm u}(k)$=$\tilde{\boldsymbol{w}}_3^{\rm u}(k)$,..., $\tilde{\boldsymbol{w}}_{n-1}^{\rm u}(k)$=$\tilde{\boldsymbol{w}}_n^{\rm u}(k)$, such that $\tilde{\boldsymbol{w}}_1^{\rm u}(k)$=...=$\tilde{\boldsymbol{w}}_n^{\rm u}(k)$ for a system of n subsystems. When the problem is distributed amongst agents, then each agent will optimise to find the local duplicate inputs. Agents then compare their local duplicate inputs to the values calculated previously by connected agents in order to achieve consensus, in the same way that agents compare other interconnecting variables.

Each agent's final $\tilde{\boldsymbol{w}}_a^{\rm u}(k)$ value will differ slightly from that of the other agents, depending on the values of c and ϵ, as these determine to what extent agents will form a consensus on variables. The control engineer must decide at the design stage which agent will ultimately decide on the value of the input to be applied to the real system being controlled, from the inputs calculated separately by each agent.

V. Simulation results

The PSO-optimised PID-based control scheme of Section III, and the centralised and distributed MPC controllers of Section IV, were used to control the multiple HVDC link system. Simulink was used to simulate the nonlinear, continuous-time power system, using the Dormand-Prince (ode45 in Matlab) continuous-time algorithm, with a maximum step size of 2ms and a relative tolerance of 0.001. The nonlinear equations used to simulate the system were based on equations (1) and (9), derived in section II-A; an example of how to derive the swing equations for generator 1 is given in Appendix A, and the power system parameters are given in Appendix B.

Linearisations of the nonlinear equations (1) and (9), were used to derive the discrete time state space models that were used in the centralised and distributed MPC controllers. These inputs were calculated and applied at fixed time steps of 10ms using Matlab and the calculated inputs were passed to the continuous time Simulink simulation. All MPC optimisations were performed using Matlab function quadprog.

Two different simulations were carried out. A 3-phase to ground fault was applied to line 1 for 100 ms after 1 ms, followed by a line break which is applied for another 100ms, and then returned to the non-fault state for the remainder of the simulation. A 3-phase to ground fault followed by a line break was similarly applied to line 3 in another simulation. All the above faults were applied to the real system but the model used for the control was based on the original non-fault scenario. All output measurements were considered noise-free.

For the simulations involving centralised controllers (the centralised MPC and PSO-optimized PID), a single control agent was used to control the whole system. However, as was stated in Section II-D, it may not be possible here to use a centralised controller as each country may have its own controller and within countries with deregulated power markets, sections of the grid may have different control agents that are responsible for the control of different sections of the grid.

For the decentralised and distributed control systems we assume here that agents 1 and 4 in Denmark are run by two separate controllers due to deregulation of that market, with one controller each for Norway and Sweden (or indeed those particular sections of the grid in those countries). Therefore, 4 different agents are responsible for these 4 different areas. In the decentralised approach the agents take a greedy approach and do not try to obtain consensus on the shared inputs between the different areas, whereas in the distributed approach the adjacent agents communicate with each other as in Fig 4.

A. Design of PID controllers using PSO

The PID parameters and bandwidths used in the controller in Fig. 3 were optimised using the PSO Toolbox [26]. The parameters used in the toolbox are given in Appendix C. Before optimisation of the controller parameters, the major oscillatory modes were found at ω_{14}=6.2118 rad/s and ω_{32}=4.0285 rad/s using eigenvalue analysis. These were then used as the centre frequencies for the bandpass filters.

50 particles were initially placed randomly across the 8-dimensional plane being optimised. Particle fitness was based on the sum of the ISE for two different faults scenarios: a 100 ms fault is applied followed by a 100 ms line break applied to lines 1 and 3 separately (the lines return to the non-fault state after the line break is finished). This ensured that the optimisation considers equally faults that happen on both sides of the HVDC links. Simulation runs lasted for 20 seconds in each case.

The optimal PID and bandwidth values found using PSO, with initial position [B_{14}, B_{23}, K_{14}, K_{23}, I_{14}, I_{23}, D_{14}, D_{23}]=[20 20 20 20 0 0 0 0], were B_{14}=2.7 rad/s, B_{23}=17.4 rad/s, K_{14}=18.9, K_{23}=460, I_{14}=3702, I_{23}=1190, D_{14}=56.2, D_{23}=0, where B_a is the filter bandwidth, K_a is the proportional gain, I_a is the integral gain, and D_a is the derivative gain used in the controller for area a.

B. Design of centralised and distributed MPC

At each sample the state equations for each generator were linearised about the current operating point as follows:

$$\begin{aligned}\frac{\rm d}{{\rm d}t}\begin{bmatrix}\delta_{{\rm r}_a}\\ \omega_{{\rm r}_a}\end{bmatrix}_{\rm op} &= \begin{bmatrix}0 & \omega_0\\ \frac{\partial f_{{\rm r}_a}}{\partial \delta_{{\rm r}_a}}|_{\rm op} & \frac{\partial f_{{\rm r}_a}}{\partial \omega_{{\rm r}_a}}|_{\rm op}\end{bmatrix}\begin{bmatrix}\delta_{{\rm r}_a}\\ \omega_{{\rm r}_a}\end{bmatrix}_{\rm op}\\ &+\begin{bmatrix}0 & 0\\ \frac{\partial f_{{\rm r}_a}}{\partial P_1^{\rm DC}}|_{\rm op} & \frac{\partial f_{{\rm r}_a}}{\partial P_2^{\rm DC}}|_{\rm op}\end{bmatrix}\begin{bmatrix}P_1^{\rm DC}\\ P_2^{\rm DC}\end{bmatrix}_{\rm op}+\begin{bmatrix}0\\ \frac{\partial f_{{\rm r}_a}}{\partial \delta_{{\rm r}_l}}|_{\rm op}\end{bmatrix}\delta_{{\rm r}_{l{\rm op}}}\end{aligned} \tag{25}$$

where in the above equation $f_{{\rm r}_a}(\delta_{{\rm r}_a},\omega_{{\rm r}_a},P_1^{\rm DC},P_2^{\rm DC},\delta_{{\rm r}_l}) = \frac{\rm d}{{\rm d}t}\omega_{{\rm r}_a}$, as defined in (9), and op indicates the linearisation of the relevant variable, vector, or function about the current operating point.

For centralised MPC the states were taken as $\boldsymbol{x}$=[δ_{r_1}, $\omega_{\mathrm{r}_1}, \ldots, \delta_{\mathrm{r}_4}$, ω_{r_4},]$^{\mathrm{T}}$, and the inputs as $\boldsymbol{u}$=[P_1^{DC} P_2^{DC}]$^{\mathrm{T}}$. For distributed MPC the states of agent a are taken as $\boldsymbol{x}_a$=[$\delta_{\mathrm{r}a}$ $\omega_{\mathrm{r}a}$]$^{\mathrm{T}}$, the inputs $\boldsymbol{u}_a$=[P_1^{DC} P_2^{DC}]$^{\mathrm{T}}$, and the interconnecting input $\boldsymbol{v}_a$=δ_{r_l}. The full system model for the centralised case was discretised using a zero-order hold with a sample time $\tau = 0.01$s, providing the discrete-time state space equations for the centralised and distributed MPC systems. Predictions in both centralised and distributed cases were formed using incremental state space models so as to ensure integral action, i.e., the augmented state $\boldsymbol{x}_{\mathrm{aug}} = [\Delta\boldsymbol{x}^{\mathrm{T}}\ \boldsymbol{x}^{\mathrm{T}}]^{\mathrm{T}}$, incremental inputs $\Delta\boldsymbol{u}$ and $\Delta\boldsymbol{u}_a$, and incremental interconnecting inputs $\Delta\boldsymbol{v}_a$ and their associated state space models are used for predictions and optimisations (these are derived as in [11]). A prediction horizon of N=50 was used so as to accurately represent the system dynamics in the optimisation.

One agent was assigned per generator to control its frequency. Each agent had access to it's relevant state space model, the constraints on it's variables, and could communicate with agents to which it was connected by an AC or HVDC link.

Each agent a's stage cost function (there is one agent for each generator, so for convenience the subscript a is used to index both), $J_a^{\mathrm{stage}}(k,p)$, for the p^{th} prediction step at sample step k, is given as follows:

$$J_a^{\mathrm{stage}}(k,p) = R_a(\omega_{\mathrm{r}_a}(k+p)-1)^2, \tag{26}$$

where R_a is the weight corresponding to ω_{r_a} in the cost function. This cost function penalises deviations of the frequency from the base frequency. The centralised MPC optimisation problem is then given by (19). The weights $[R_1, \ldots, R_4]$=[10 30 10 10] were used for both MPC cases.

The interconnection cost for the distributed MPC case at sample step k and iteration l of the control cycle, $J_a^{\mathrm{inter}}(k,l)$, is formed from a hypothetical centralised augmented Lagrangian MPC formulation which is given as follows:

$$\min_{\Delta\tilde{\boldsymbol{u}}_1,\ldots,\Delta\tilde{\boldsymbol{u}}_4} \sum_{a=1}^{4}\left(J_a^{\mathrm{local}}\right)+ \begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{41}^{\mathrm{in},\boldsymbol{x}_4} \\ \tilde{\boldsymbol{\lambda}}_{32}^{\mathrm{in},\boldsymbol{x}_3} \\ \tilde{\boldsymbol{\lambda}}_{23}^{\mathrm{in},\boldsymbol{x}_2} \\ \tilde{\boldsymbol{\lambda}}_{14}^{\mathrm{in},\boldsymbol{x}_1} \\ \tilde{\boldsymbol{\lambda}}_{41}^{\mathrm{u}} \\ \tilde{\boldsymbol{\lambda}}_{12}^{\mathrm{u}} \\ \tilde{\boldsymbol{\lambda}}_{23}^{\mathrm{u}} \\ \tilde{\boldsymbol{\lambda}}_{34}^{\mathrm{u}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \tilde{\boldsymbol{w}}_{41}^{\mathrm{in},\boldsymbol{x}_4}-\tilde{\boldsymbol{w}}_{14}^{\mathrm{out},\boldsymbol{x}_4} \\ \tilde{\boldsymbol{w}}_{32}^{\mathrm{in},\boldsymbol{x}_3}-\tilde{\boldsymbol{w}}_{23}^{\mathrm{out},\boldsymbol{x}_3} \\ \tilde{\boldsymbol{w}}_{23}^{\mathrm{in},\boldsymbol{x}_2}-\tilde{\boldsymbol{w}}_{32}^{\mathrm{out},\boldsymbol{x}_2} \\ \tilde{\boldsymbol{w}}_{14}^{\mathrm{in},\boldsymbol{x}_1}-\tilde{\boldsymbol{w}}_{41}^{\mathrm{out},\boldsymbol{x}_1} \\ \tilde{\boldsymbol{w}}_1^{\mathrm{u}}-\tilde{\boldsymbol{w}}_4^{\mathrm{u}} \\ \tilde{\boldsymbol{w}}_2^{\mathrm{u}}-\tilde{\boldsymbol{w}}_1^{\mathrm{u}} \\ \tilde{\boldsymbol{w}}_3^{\mathrm{u}}-\tilde{\boldsymbol{w}}_2^{\mathrm{u}} \\ \tilde{\boldsymbol{w}}_4^{\mathrm{u}}-\tilde{\boldsymbol{w}}_3^{\mathrm{u}} \end{bmatrix} + \frac{c}{2}\left\| \begin{matrix} \tilde{\boldsymbol{w}}_{41}^{\mathrm{in},\boldsymbol{x}_4}-\tilde{\boldsymbol{w}}_{14}^{\mathrm{out},\boldsymbol{x}_4} \\ \tilde{\boldsymbol{w}}_{32}^{\mathrm{in},\boldsymbol{x}_3}-\tilde{\boldsymbol{w}}_{23}^{\mathrm{out},\boldsymbol{x}_3} \\ \tilde{\boldsymbol{w}}_{23}^{\mathrm{in},\boldsymbol{x}_2}-\tilde{\boldsymbol{w}}_{32}^{\mathrm{out},\boldsymbol{x}_2} \\ \tilde{\boldsymbol{w}}_{14}^{\mathrm{in},\boldsymbol{x}_1}-\tilde{\boldsymbol{w}}_{41}^{\mathrm{out},\boldsymbol{x}_1} \\ \tilde{\boldsymbol{w}}_1^{\mathrm{u}}-\tilde{\boldsymbol{w}}_4^{\mathrm{u}} \\ \tilde{\boldsymbol{w}}_2^{\mathrm{u}}-\tilde{\boldsymbol{w}}_1^{\mathrm{u}} \\ \tilde{\boldsymbol{w}}_3^{\mathrm{u}}-\tilde{\boldsymbol{w}}_2^{\mathrm{u}} \\ \tilde{\boldsymbol{w}}_4^{\mathrm{u}}-\tilde{\boldsymbol{w}}_3^{\mathrm{u}} \end{matrix} \right\|_2^2, \tag{27}$$

where the ks and ls, used to denote the sample step and distributed MPC iteration, are omitted for compactness. This formulation enables the distribution of the problem so that agents can reach agreement on the control inputs, i.e., the HVDC powers.

Each agent a has a duplicate vector of the control inputs $\tilde{\boldsymbol{w}}_a^{\mathrm{u}}(k)$ where $\boldsymbol{w}_a^{\mathrm{u}}(k) = [P_1^{\mathrm{DC}}(k)\ P_2^{\mathrm{DC}}(k)]^{\mathrm{T}}$. The order in which agents optimise for the distributed MPC cycles starts with agent 1 and ends with 4. Therefore in the hypothetical centralised augmented Lagrangian case, the equality constraint $\tilde{\boldsymbol{w}}_a^{\mathrm{u}}(k) = \tilde{\boldsymbol{w}}_{a,\mathrm{last}}^{\mathrm{u}}(k)$ is applied for each agent ($w_{a,\mathrm{last}}^{\mathrm{u}}$ denotes the last agent to optimise) in order to reach consensus on the duplicate input values. Interconnecting constraints between interconnecting state variables are also applied.

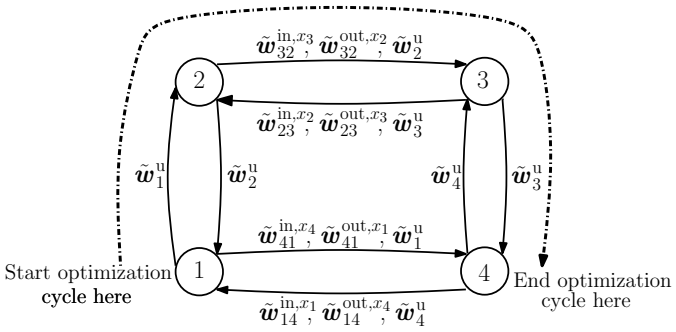


Fig. 4. Order of serial distributed MPC optimisations and variables communicated between agents

When (27) is distributed amongst the agents, $J_a^{\mathrm{inter}}(k,l)$ takes the following distributed form for agent a, where bus j is AC-connected to bus a:

$$J_a^{\mathrm{inter}} = \begin{bmatrix} \tilde{\boldsymbol{\lambda}}_{ja}^{\mathrm{in},\boldsymbol{x}_j} \\ -\tilde{\boldsymbol{\lambda}}_{aj}^{\mathrm{in},\boldsymbol{x}_a} \\ \tilde{\boldsymbol{\lambda}}_a^{\mathrm{u}} \\ -\tilde{\boldsymbol{\lambda}}_{a,\mathrm{next}}^{\mathrm{u}} \end{bmatrix}^{\mathrm{T}} \begin{bmatrix} \tilde{\boldsymbol{w}}_{ja}^{\mathrm{in},\boldsymbol{x}_j} \\ \tilde{\boldsymbol{w}}_{ja}^{\mathrm{out},\boldsymbol{x}_a} \\ \tilde{\boldsymbol{w}}_a^{\mathrm{u}} \\ \tilde{\boldsymbol{w}}_a^{\mathrm{u}} \end{bmatrix} + \frac{c}{2}\left\| \begin{matrix} \tilde{\boldsymbol{w}}_{aj,\mathrm{prev}}^{\mathrm{out},\delta_{\mathrm{r}_j}} - \tilde{\boldsymbol{w}}_{ja}^{\mathrm{in},\boldsymbol{x}_j} \\ \tilde{\boldsymbol{w}}_{aj,\mathrm{prev}}^{\mathrm{in},\boldsymbol{x}_a} - \tilde{\boldsymbol{w}}_{ja}^{\mathrm{out},\boldsymbol{x}_a} \\ \tilde{\boldsymbol{w}}_{\mathrm{last,prev}}^{\mathrm{u}} - \tilde{\boldsymbol{w}}_a^{\mathrm{u}} \\ \tilde{\boldsymbol{w}}_{\mathrm{next,prev}}^{\mathrm{u}} - \tilde{\boldsymbol{w}}_a^{\mathrm{u}} \end{matrix} \right\|_2^2, \tag{28}$$

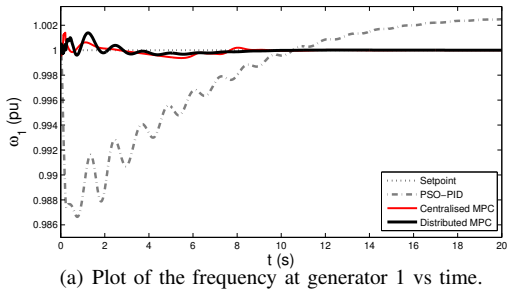
where $w_{a,\mathrm{next}}^{\mathrm{u}}$ denotes the next agent to optimise and the ks and ls, used to denote the sample step and distributed MPC iteration, are dropped for compactness.

After agent a has completed its optimisation, it sends the relevant updated values of the variables to the agents that are connected to it, for use in their distributed MPC optimisations. The total cost function for agent a is given by (21). This can be put into quadratic form using simple matrix manipulation, where the optimisation vector is $\Delta\tilde{\boldsymbol{u}}_{opt}(k) = [\Delta\tilde{\boldsymbol{u}}^{\mathrm{T}}(k)\ \Delta\tilde{\boldsymbol{w}}_{\mathrm{in}}^{\mathrm{T}}(k)]^{\mathrm{T}}$.

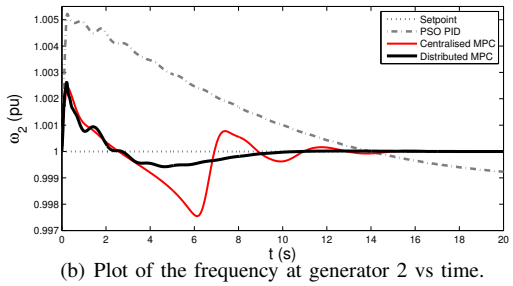
The HVDC link ranges are $-2 \leq P^{\mathrm{DC}}(k) \leq 2$ pu and the frequency range at all buses is $0.984 \leq \omega(k) \leq 1.016$ pu. These constraints are applied over the full prediction horizon. The distributed MPC parameters related to communication are given as follows: $c = 0.1$, $\epsilon = 10^{-2}$.

In the centralised MPC case, the optimal values calculated for $P_1^{\mathrm{DC}}(k)$ and $P_2^{\mathrm{DC}}(k)$ are applied to the system. The 4 agents in the distributed MPC system calculate slightly different values for the HVDC powers to each other, as these powers only have to match to a degree, determined by the distributed MPC parameters c and ϵ. Therefore, one agent per HVDC link a is assigned to apply its calculated $P_a^{\mathrm{DC}}(k)$ value to the system at sample k.

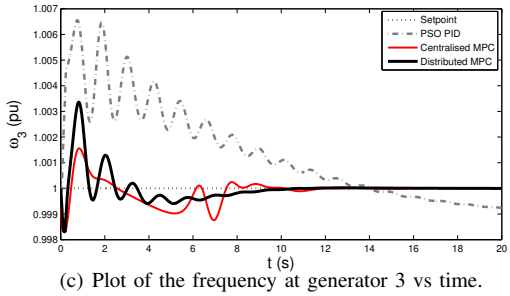
Here the values for $P_1^{\mathrm{DC}}(k)$ and $P_2^{\mathrm{DC}}(k)$, calculated by agents 2 and 3 respectively, are the control inputs that are applied (these were chosen as the vast majority of power transfer is from agents 2 and 3 to agents 1 and 4, and so



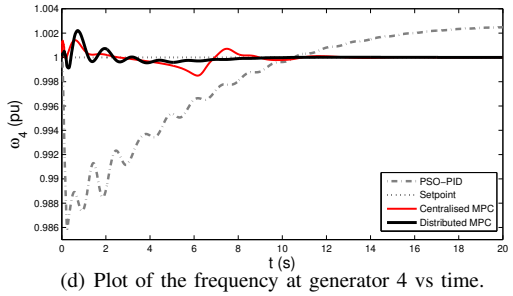
(a) Plot of the frequency at generator 1 vs time.



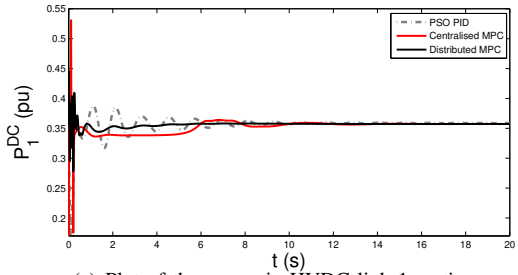
(b) Plot of the frequency at generator 2 vs time.



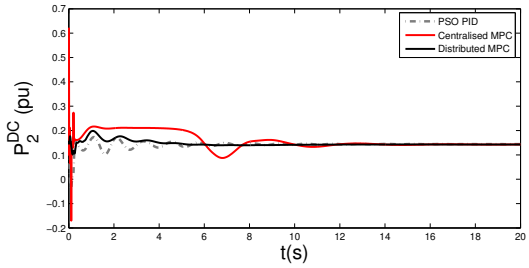
(c) Plot of the frequency at generator 3 vs time.



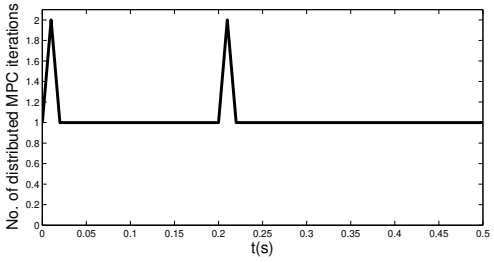
(d) Plot of the frequency at generator 4 vs time.



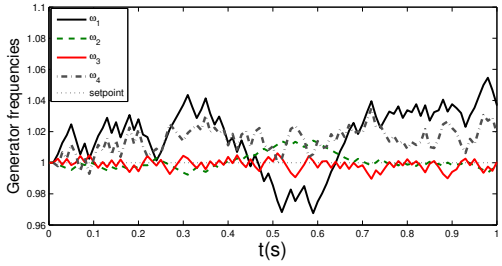
(e) Plot of the power in HVDC link 1 vs time.



(f) Plot of the power in HVDC link 2 vs time.



(g) Plot of the first 0.5 seconds of distributed MPC iterations vs time (iterations stay at 1 for remainder of simulation).



(h) Plot of the generator frequencies vs time for decentralised MPC (simulation stopped at t=1 s due to constraints violations).

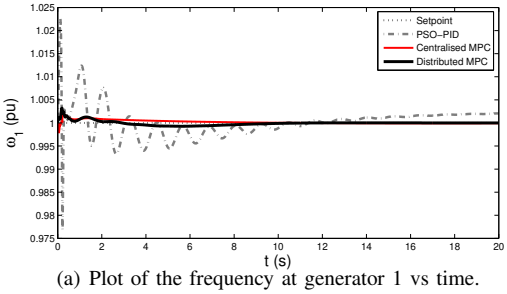
Fig. 5. Plots of pu frequency and HVDC powers vs. time for a 100ms fault followed by a 100ms line break applied to line 1.

it is assumed these agents insist on having the final say on what power is allowed to be transferred to agents 1 and 4).

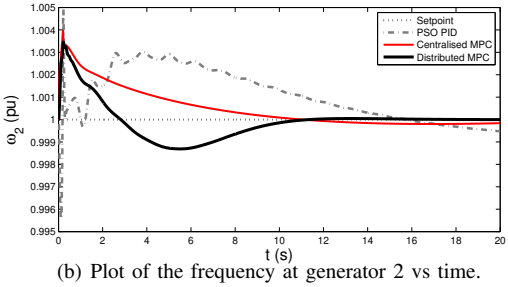
C. Results

Two simulations were run in which an AC line was subjected to a 100ms line fault, followed by 100ms line break, and then returned to its original non-fault state (a 100ms line break is applied here as it is the longest line break for which the performances of both MPC and PID controllers are comparable. Line breaks longer than this tended to result in instability for the system under PID control). In the first simulation this fault scenario was applied to line 1, and it was applied to line 3 in the second simulation. The results of the simulations can be seen in Figs. 5 and 6, which show the frequencies of each generator, the applied HVDC powers, and the number of distributed MPC iterations needed at each sample step, plotted against time for each control system, for the first and second simulations, respectively.

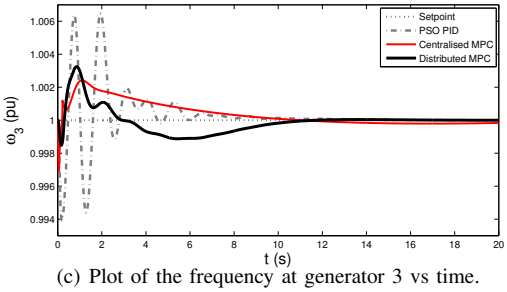
First of all it can be seen that the decentralised control gives by far the worst performance (simulations are terminated due to excessive constraint violations) and that at least some level of communication is necessary between agents to control this system. This becomes apparent from Fig. 6(h). Agents 2 and 3 are responsible for applying the final control inputs to the system and so are not affected by the line fault on line 3 and hence these agents remain at 1 pu. Therefore they do not



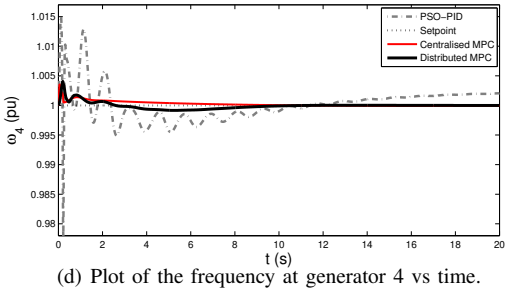
(a) Plot of the frequency at generator 1 vs time.



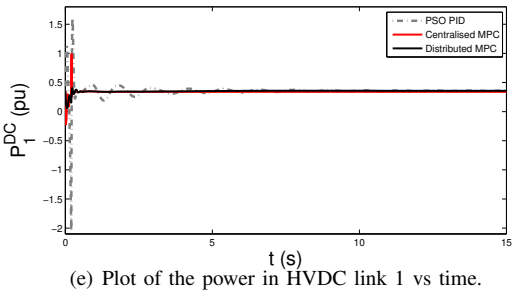
(b) Plot of the frequency at generator 2 vs time.



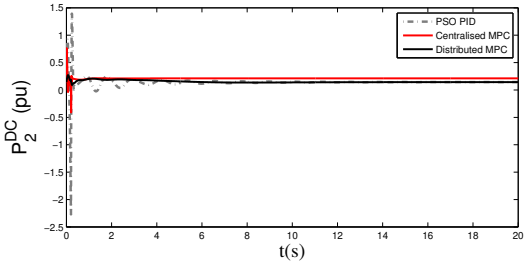
(c) Plot of the frequency at generator 3 vs time.



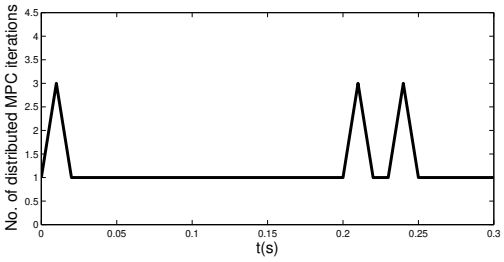
(d) Plot of the frequency at generator 4 vs time.



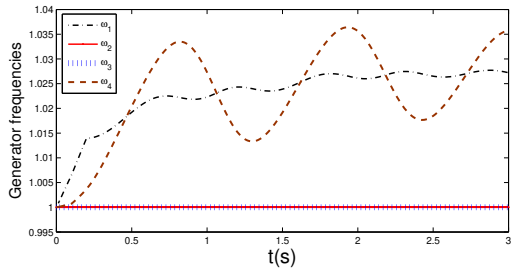
(e) Plot of the power in HVDC link 1 vs time.



(f) Plot of the power in HVDC link 2 vs time.



(g) Plot of the first 0.3 seconds of distributed MPC iterations vs time (iterations stay at 1 for remainder of simulation).



(h) Plot of the generator frequencies vs time for decentralised MPC (simulation stopped at t=3 s due to constraints violations).

Fig. 6. Plots of pu frequency and HVDC powers vs. time for a 100ms fault followed by a 100ms line break applied to line 3.

change the control inputs to help stabilise the frequencies of generators 1 and 4. However, even in Fig. 5(h) when they are affected by the line fault in line 1 they are not able to satisfactorily restabilise the system.

It can be seen in Table I that the distributed MPC yields the best performance in an ISE sense, followed by the centralised MPC, and finally the PSO-optimised PID performance scheme. The MPC strategies do not experience the large unacceptable deviations from the setpoint experienced by the PID controller in Fig. 6, which violate the constraints on the maximum allowable frequency deviations. The MPC strategies can also be seen to have improved the system damping. Of note, in the presented scenario, is the fact that due to limited horizons and discrepancies between the real world and MPC models, the distributed MPC performs outperforms the centralised one.

In general, the techniques developed in [21] give acceptable performance for fault scenarios of less than 100ms in duration. However, at least in the case of the PID controller, the performance of these controllers can become unacceptable in the face of more serious faults.

The trade-off experienced by the centralised and distributed MPC controllers for better disturbance rejection over the PID control scheme is a significant computational overhead in both cases, and a communications overhead in the case of the distributed MPC. The average and longest times taken to

Fault on Line	PSO-PID	MPC	
		Centralised	Distributed
1	0.1161	0.0022	0.0015
3	0.0582	0.0042	0.0037
Total	0.1743	0.0066	0.0052

TABLE I
COMPARISON OF ISE FOR PSO-PID AND MPC SCHEMES

compute the control inputs for a centralised MPC cycle were 0.47 s and 1.125 s, respectively, and the average and longest times taken to converge on final solution for a distributed MPC cycle were 0.82 s and 1.8 s, respectively, on a computer with an Intel® Core™ 2 6400 operating at 2.13 GHz and with 3 GB of RAM (These times were taken as the time from the linearisation of the state space to the application of the control inputs, measured using the cputime command in Matlab. The actual time for a multi-core processor is roughly equal to the cputime divided by the number of cores). It should be noted that the distributed MPC simulation was also performed on an single PC, rather than several different PCs as would be the case for a real implementation.

It can be seen from these results that the computational effort needed for the distributed MPC problem is larger than that of the centralised MPC, as well as having an added communications overhead. The number of distributed MPC iterations necessary to complete each optimisation cycle at each sample in each simulation, which represents the level of communication necessary, are given in Figs. 5(g) and 6(g). However, this centralised MPC problem is still relatively small. For larger power systems the centralised MPC problem would become increasingly intractable computationally, whereas the distributed MPC problems would stay the same size. However, for distributed MPC problems the amount of communication necessary between agents would increase with the size of the problem. Also, it should again be noted that there are situations such as those depicted in this paper, where a multi-agent approach is desirable due to a number of separate controllers controlling different subsystems, where it may not be possible to adopt a centralised control approach.

With regards to disturbance rejection and stability performance in a larger power network one could expect that for disturbances of a similar size the distributed MPC would continue to provide satisfactory control. However, it is possible that with larger disturbances, a larger number of agents, as well as the added factor of a limited decision making time (which would impact the potential number of iterations per distributed MPC cycle), there would be some performance degradation. The same would be expected of centralised MPC, though.

In the example in this paper, generation capacities are kept constant and the modulation of the HVDC links alone is used to restabilise the system. This system is therefore a useful testbed for demonstrating the capabilities of HVDC links alone to stabilise systems. System performance could potentially be further improved by taking into account varying generator capacities.

VI. CONCLUSIONS

Here the applications of a Particle Swarm Optimisation (PSO) optimised PID controller, and a centralised and distributed Model Predictive Control (MPC) controller to a multiple High Voltage Direct Current (HVDC) link system have been discussed. The distributed MPC gives the best result, followed by the centralised MPC, and then the PSO optimised PID controller.

It has been seen that decentralised control is highly unsuitable for the control of this system. As centralised MPC problems can get quite large for power systems, and given the improvement in performance associated with distributed MPC over the PSO PID controller, it can be seen that in large power systems distributed MPC is an attractive option for advanced control.

There is much potential for further research using this system as a benchmark, such as the need to further reduce the computational and communication overhead associated with the distributed MPC technique used here in order to make practical application more feasible. Furthermore, stability and convergence guarantees should be investigated for this technique. Communication delays and data transmission errors are other issues that would affect the control performance. It would be interesting to investigate how the feedback linearisation controller in [21] deals with faults of the magnitude of those studied in this paper.

It would also be interesting to assess the performance of robust control techniques on this system. Parametric uncertainties could be included, and more complex system dynamics could be added to the HVDC lines, and generators, for example, to make the system more realistic for the application of such techniques. Also, it would be interesting to see how controller performance is affected when the system is of a larger scale.

ACKNOWLEDGEMENTS

The authors would like to thank the reviewers and editor of the journal for their constructive comments. The authors would also like to thank Dr. Mick Egan of the Department of Electrical and Electronic Engineering, University College Cork, Ireland for his help regarding power systems theory. This work was funded by the Irish Research Council for Science, Engineering and Technology (IRCSET) and supported by the BSIK project "Next Generation Infrastructures (NGI)", the Delft Research Centre Next Generation Infrastructures, the European STREP project "Hierarchical and distributed model predictive control (HD-MPC)", contract number INFSO-ICT-223854, and the VENI project "Intelligent multi-agent control for flexible coordination of transport hubs" (project 11210) of the Dutch Technology Foundation STW.

REFERENCES

[1] P. Kundur, *Power System Stability and Control*. McGraw Hill, New York, 1994.

[2] R. Eriksson and V. Knazkins, "On the coordinated control of multiple HVDC links," in *Transmission and Distribution Conference and Exposition: Latin America, 2008 IEEE/PES*, Aug. 2008, pp. 1–6.

[3] R. Eriksson and V. Knazkin, "Nonlinear coordinated control of multiple HVDC links," *IEEE 2nd International*

Power and Energy Conference PECon 2008, pp. 497–501, Dec. 2008.

[4] D. Kwok and F. Sheng, "Genetic algorithm and simulated annealing for optimal robot arm PID control," in *Proceedings of the IEEE World Congress on Computational Intelligence*, vol. 2, June 1994, pp. 707–713.

[5] A. Jones and P. De Moura Oliveira, "Genetic auto-tuning of PID controllers," *First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pp. 141–145, Sept. 1995.

[6] Z.-L. Gaing, "A Particle Swarm Optimization approach for optimum design of PID controller in AVR system," *IEEE Transactions on Energy Conversion*, vol. 19, no. 2, pp. 384–391, June 2004.

[7] P. Fabijanski and R. Lagoda, "On-line PID controller tuning using genetic algorithm and DSP PC board," in *13th Power Electronics and Motion Control Conference*, Sept. 2008, pp. 2087–2090.

[8] G. Lin and G. Liu, "Tuning PID controller using adaptive genetic algorithms," in *5th International Conference on Computer Science and Education (ICCSE)*, Aug. 2010, pp. 519–523.

[9] J. Maciejowski, *Predictive Control with Constraints*. Harlow, England: Prentice Hall, 2002.

[10] J. Rawlings and D. Mayne, *Model Predictive Control: Theory and Design*. Madison, Wisconsin: Nob Hill Publishing, 2009.

[11] L. Wang, *Model Predictive Control System Design and Implementation Using Matlab*. Springer, London, 2009.

[12] J. Rossiter, *Model Based Predictive Control-A Practical Approach*. CRC Press, Florida, 2003.

[13] R. Scattolini, "Architectures for distributed and hierarchical Model Predictive Control - A review," *Journal of Process Control*, vol. 19, no. 5, pp. 723–731, 2009.

[14] G. Sanchez, L. Giovanini, M. Murillo, and A. Limache, "Distributed model predictive control based on dynamic games," *Advanced Model Predictive Control*, 2011.

[15] J. Liu, X. Chen, D. Muñoz de la Peña, and P. D. Christofides, "Sequential and iterative architectures for distributed model predictive control of nonlinear process systems," *AIChE Journal*, vol. 56, no. 8, pp. 2137–2149, 2010.

[16] J. Liu, D. Muñoz de la Peña, and P. Christofides, "Distributed Model Predictive Control of Nonlinear Process Systems," *AIChE Journal*, vol. 55, pp. 1171–1184, 2009.

[17] R. R. Negenborn, B. De Schutter, and J. Hellendoorn, "Multi-agent model predictive control for transportation networks: Serial versus parallel schemes," *Engineering Applications of Artificial Intelligence*, vol. 21, no. 3, pp. 353–366, April 2008.

[18] A. Venkat, "Distributed Model Predictive Control: Theory and Applications," Ph.D. dissertation, University of Wisconsin-Madison, Wisconsin, 2006.

[19] R. R. Negenborn, G. Hug-Glanzmann, B. De Schutter, and G. Andersson, "A novel coordination strategy for multi-agent control using overlapping subnetworks with application to power systems," in *Efficient Modeling and Control of Large-Scale Systems*, J. Mohammadpour and K. M. Grigoriadis, Eds. Norwell, Massachusetts: Springer, 2010, pp. 251–278.

[20] S. Talukdar, D. Jia, P. Hines, and B. Krogh, "Distributed Model Predictive Control for the Mitigation of Cascading Failures," in *Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference*, Seville, Spain, Dec. 2005, pp. 4440–4445.

[21] R. Erikkson, "Security-centered coordinated control in AC/DC transmission systems," Licentiate Thesis, Royal Institute of Technology, School of Electrical Engineering, Electric Power Systems, Stockholm, Sweden, 2008.

[22] M. Pai, K. Padiyar, and C. Radhakrishna, "Transient stability analysis of multi-machine AC/DC power systems via energy-function method," *IEEE Transactions on Power Apparatus and Systems*, vol. 100, no. 12, pp. 5027–5035, Dec. 1981.

[23] UCTE, "Policy 1: Load-frequency control and performance," *UCTE operation handbook*, p. 2, 2004. [Online]. Available: http://www.pse-operator.pl/uploads/kontener/UCTE_Operation_Handbook_Policy_1.pdf

[24] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, 1995, pp. 1942–1948.

[25] I. C. Trelea, "The Particle Swarm Optimization algorithm: convergence analysis and parameter selection," *Information Processing Letters*, vol. 85, pp. 317–325, 2003.

[26] B. Birge, "PSOt - a Particle Swarm Optimization toolbox for use with Matlab," *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, pp. 182–186, April 2003.

APPENDIX A
CALCULATING SWING PARAMETERS FOR GENERATOR 1

For the multiple HVDC link system in Fig. 1 the $\boldsymbol{G}$, $\boldsymbol{B}$, and $\boldsymbol{Y}_{\mathrm{DC}}$ matrices are formed. The elements of $\boldsymbol{Y}_{\mathrm{DC}}$ are given by $\vec{d}_{i,j} = d^{\mathrm{R}}_{i,j} + \mathrm{j}d^{\mathrm{I}}_{i,j}$. Taking equation (9) for generator 1 gives:

$$\begin{aligned}\dot{\omega}_{\mathrm{r}_1} = \frac{1}{2H_1}\Big(&P_{\mathrm{m}_1} - G_{1,1}E'^{2}_{\mathrm{q}_1} - \\ &E'_{\mathrm{q}_1}E'_{\mathrm{q}_4}(G_{1,4}\cos(\delta_{\mathrm{r}_1}-\delta_{\mathrm{r}_4}) + B_{1,4}\sin(\delta_{\mathrm{r}_1}-\delta_{\mathrm{r}_4})) \\ &+ g_{1,1}P^{\mathrm{DC}}_1 + g_{1,2}P^{\mathrm{DC}}_2 - D_1\Delta\omega_{r_1}\Big)\end{aligned} \tag{29}$$

The $g_{1,1}$ and $g_{1,2}$ parameters are derived as follows: Assume that power flows from bus 2 to 1 in HVDC link 1 and from bus 3 to 4 in HVDC link 2:

$$\boldsymbol{I}_{\mathrm{DC}} = \begin{pmatrix}\vec{I}^{\mathrm{DC}}_{1,1}\\ \vec{I}^{\mathrm{DC}}_{2,1}\\ \vec{I}^{\mathrm{DC}}_{3,2}\\ \vec{I}^{\mathrm{DC}}_{4,2}\end{pmatrix} = \begin{pmatrix}(\frac{P^{\mathrm{DC}}_1 - \mathrm{j}Q^{\mathrm{DC}}_1}{\vec{U}_1})^*\\ (\frac{-P^{\mathrm{DC}}_1 - \mathrm{j}Q^{\mathrm{DC}}_1}{\vec{U}_2})^*\\ (\frac{P^{\mathrm{DC}}_2 - \mathrm{j}Q^{\mathrm{DC}}_2}{\vec{U}_3})^*\\ (\frac{-P^{\mathrm{DC}}_2 - \mathrm{j}Q^{\mathrm{DC}}_2}{\vec{U}_4})^*\end{pmatrix}$$

The generator power

$$P_{\mathrm{G}_1} = \mathbb{R}(\vec{E}_{\mathrm{q}_1}\vec{I}^{*}_{\mathrm{g}_1})$$

$$=\mathbb{R}\left(\vec{E}_{\mathrm{q}_1}\boldsymbol{T}_{\vec{I}_{\mathrm{q}_1}}\left((\boldsymbol{G}+\mathrm{j}\boldsymbol{B})\boldsymbol{E}+\boldsymbol{Y}_{\mathrm{DC}}\boldsymbol{I}_{\mathrm{DC}}\right)^*\right)$$

where $\boldsymbol{T}_{\vec{I}_{\mathrm{G}_1}}$ is a matrix of ones that picks out $\vec{I}_{\mathrm{G}_1}$. It is the $\boldsymbol{Y}_{\mathrm{DC}}\boldsymbol{I}_{\mathrm{DC}}$ part of this term that gives the $g_{1,1}$ and $g_{1,2}$ parameters:

$$\begin{aligned}&g_{1,1}P_1^{\mathrm{DC}}+g_{1,2}P_2^{\mathrm{DC}}\\ =&\mathbb{R}\Big(-\vec{E}_{\mathrm{G}_1}T_{\vec{I_{\mathrm{G}_1}}}(\boldsymbol{Y}_{\mathrm{DC}}\boldsymbol{I}_{\mathrm{DC}})^*\Big)\\ =&\mathbb{R}\Big(-\tfrac{\vec{E}_{\mathrm{G}_1}}{\vec{U}_1}(d_{1,1}^{\mathrm{R}}+\mathrm{j}d_{1,1}^{\mathrm{I}})^*(P_1^{\mathrm{DC}}(1-\mathrm{j}q_{\mathrm{r}_1}))\\ &-\tfrac{\vec{E}_{\mathrm{G}_1}}{\vec{U}_4}(d_{1,4}^{\mathrm{R}}+\mathrm{j}d_{1,4}^{\mathrm{I}})^*(P_2^{\mathrm{DC}}(1-\mathrm{j}q_{\mathrm{r}_2}))\Big)\end{aligned}$$

where q_{r_1} and q_{r_2} are the ratios of reactive to active power in HVDC links 1 and 2 respectively and $\delta_{\mathrm{r}_a,0}$ and θ_{j_0} denote the initial conditions of the rotor angle of generator a and the bus angle at bus j respectively. It should also be noted that the complex $\frac{\vec{E}}{\vec{U}}$ ratios are taken as constant in order to simplify the equations. This process is then repeated for each generator.

APPENDIX B
POWER SYSTEM PARAMETERS USED IN THE SIMULATION

$S_{\mathrm{base}}=100\times10^6$ VA, $U_{\mathrm{base}}=100\times10^3$ V, $f_{\mathrm{base}}=50$ Hz, $w_0=2\pi f_{\mathrm{base}}$ rad/s.

Line	1	2	3	4
X_{L} pu	0.6	0.6	0.1	0.1
X_{S} pu	0.1	0.1	0.1	0.1

Generator	1	2	3	4
$x_{\mathrm{d}}^{'}$ pu	0.09	0.06	0.12	0.12
H (s)	2	4	2	2
D pu	1	1	1	1
$P_{\mathrm{G}}=P_{\mathrm{m}}$ pu	0.1	0.6	0.1	0.1
δ_{r_0} rad	5.9874	0.2871	5.585	5.03
$E_{\mathrm{q}}^{'}$ pu	0.4454	0.513	0.6807	1.0622

Bus	1	2	3	4
Load pu	0.1+0.05i	0.1+0.05i	0.1+0.05i	0.6+0.2759i
U pu	0.1097	0.2426	0.256	0.2219
θ rad	-0.4809	6.2768	5.5161	-1.3042

HVDC link a=	1	2
$P_{a,0}^{\mathrm{DC}}$ pu	0.3573	0.1427
q_{r_a}	0.8952	0.9037

APPENDIX C
PSO TOOLBOX PARAMETERS

The bounds for the optimisation of B, K_P, K_I, K_D are: $0.01{\leq}B_{14}{\leq}20$, $0.01{\leq}B_{32}{\leq}20$, $0{\leq}K_{\mathrm{P}}^{14}{\leq}1000$, $0{\leq}K_{\mathrm{P}}^{32}{\leq}1000$, $0{\leq}K_{\mathrm{I}}^{14}{\leq}10000$, $0{\leq}K_{\mathrm{I}}^{32}{\leq}10000$, $0{\leq}K_{\mathrm{D}}^{14}{\leq}1000$, $0{\leq}K_{\mathrm{D}}^{32}{\leq}1000$. Parameters used for the PSO Toolbox are given as follows:

Parameter	Description	
p	number of particles	50
mvden	max. velocity divisor	2
errgrad	error gradient tolerance	1e-5
epoch	maximum number of iterations	2000
errgraditer	number of epochs without errgrad change before termination	9