Real-Time Dynamic Optimization-based Advisory System for Electric Arc Furnace Operation

Smriti Shyamal and Christopher L.E. Swartz*

Department of Chemical Engineering, McMaster University, 1280 Main Street West, Hamilton, Ontario, Canada, L8S 4L7

E-mail: swartzc@mcmaster.ca

Abstract

Electric arc furnaces (EAFs) are widely applied in the steel industry for producing steel by melting scrap metal. This highly energy intensive steelmaking process is subject to limited automation, with decisions related to input amounts and timings typically made by operators. This leads to suboptimal EAF batch operation due to complex behavior and relationships between variables that are inevitably overlooked in the decision making. In this work, we introduce an advisory system that employs a first-principles EAF model to support the operator decision making in real time for economically optimal process operation. A dynamic optimization calculation can be triggered by the operator at any point in the batch, an action that can be repeated multiple times during the batch. The advisory system incorporates a multi-rate moving horizon estimator (MHE) that continually computes estimates of the process states utilizing current and past inputs and measurements. End-point constraints and potential

*To whom correspondence should be addressed

extension of the batch duration are handled through a multi-tiered optimization algorithm. Case studies demonstrate a major economic improvement when the dynamic optimization-based advisory system is used. We show that the online computational load is under 5 seconds on average when a proposed multi-tiered initialization scheme is used for solving the large-scale optimal control problems.

1 Introduction

Electric arc furnaces (EAFs) are widely used in the steel industry for production of steel by melting scrap metal collected from different sources, with approximately 25% of the world's steel production generated using EAFs.[1] Melting of steel is a highly energy intensive batch process, with the required energy provided in form of electrical and chemical energy. A single batch operation consumes approximately 400 kilowatt-hours/ton of steel produced.[2] The electrical energy is supplied through electric arcs to the scrap metal via multiple electrodes. The main sources of chemical energy are combustion of natural gas and oxygen injected through burners, and exothermic chemical reactions in a slag layer that floats above the molten metal. Also added to the EAF are carbon (both directly charged and lanced), fluxes comprising lime and dolomite for formation of a reactive slag, and lanced oxygen. Key reactions taking place are formation of CO that contributes to a foamy slag that is useful for insulation of the furnace walls and roof from the arc, and oxidation of several elements in the bath. The high energy consumption of EAFs motivates the research and development of advanced control, estimation and optimization strategies for efficient and cost optimal operation. However, large fluctuations during the process, harsh operating conditions, and the limited availability of measurements pose challenges for real-time process optimization. EAFs are currently operated without the use of sophisticated optimization tools, with EAF operating decisions typically made based on what has worked in the past. Given the relatively low level of automation in EAFs, fully automated dynamic real-time optimization is unlikely

to be accepted by EAF operating personnel at present. However, there is an opportunity for designing an advisory system where operators have a choice as to whether or not to implement the control moves calculated by economics-based real-time dynamic optimization. In this work, we introduce a real-time advisory system to enhance the EAF operation and generate savings through optimal utilization of inputs.

Early research works on EAF optimization focused on finding optimal operating decisions using a simplified model, or a model describing only a subprocess of the EAF system. Woodside et al.[3] utilized a 2 state model to formulate an optimal control problem to minimize EAF power usage. Oosthuizen et al.[4] formulated and solved a set-point tracking model predictive control (MPC) problem which utilized a linearized dynamic model. Bekker et al.[5] considered a two-input-three-output MPC configuration for controlling an EAF off-gas system. While a linear dynamic model may yield adequate performance for set-point tracking in EAF operations, such models are less suitable for economic optimization due to model inaccuracy over the typically wide operating range. Matson and Ramirez[6] developed a full-scale EAF model based on dynamic mass and energy balances, and used it to conduct dynamic optimization to minimize the FeO production, batch time and offgas CO content. More recently, MacRosty and Swartz[7] developed a first-principles based differential-algebraic equation (DAE) model for an EAF, and subsequently used it in a rigorous dynamic optimization framework.[8] They showed that dynamic optimization can be used to maximize profit by considering trade-offs between various inputs and processing time. The approach was extended in MacRosty and Swartz[9] to an economics-based nonlinear MPC (NMPC) framework for EAF operation. However, full noise-free state measurements were assumed to be available, and a sequential solution approach was applied for the solution of the NMPC dynamic optimization problems. Rashid et al.[10,11] formulated and solved an economic model predictive control (EMPC) problem for EAF operation, but used data-driven models instead of a rigorous first-principles model. Multi-rate moving horizon estimation (MHE) in conjunction with NMPC was recently used to design a real-time energy management strategy for EAF operation.[12] The

optimization decisions were limited to five inputs, and no strategy was provided to deal with potential infeasibilities associated with the end-point constraints. Our focus in this article is on the development of a framework for operator-trigged reoptimization, with the explicit handling of potential end-point infeasibilities through a multi-tiered optimization strategy.

For on-line reoptimization, state knowledge is necessary for initialization of the dynamic optimization problem. Although many tools such as the extended Kalman filter (EKF),[13] unscented Kalman filter (UKF),[14] or particle filter are used by researchers for state estimation, MHE has become popular over the last decade due to its straightforward way of handling constraints,[15–17] use of a full-scale dynamic model in optimization and utilization of state-of-the-art numerical optimization algorithms for fast solution times.[18] Significant effort is usually directed to solving the MHE problems quickly so as to avoid feedback delays. Since EAFs are characterized by limited availability of sensor measurements, it is difficult to obtain quality state estimates for an on-line implementation. Moreover, the measurements have different sampling rates and unknown disturbances often affect the process operation. There are limited research works available where state estimation is applied to the EAF. An EKF was employed by Billings and coworkers[19] for estimating states when the EAF process is in the refining stage. Since the model had only 4 states, it is insufficient to capture the detailed process complexity. A constrained multi-rate EKF was also employed using a linearized state space model of the EAF.[20] An MHE optimization formulation for solution via a sequential approach was proposed by Shyamal and Swartz.[21]

In this article, we propose a real-time dynamic optimization-based advisory system for EAF process operation. The advisory system utilizes a first-principles model to provide decision support to plant operators by computing economics-based optimal input profiles. It uses multi-rate MHE to calculate state estimates at every sample time; however, they are sent to the multi-tiered optimizer only if the operator calls for decision support. The operator-triggered framework is well-suited for primary operations in the steel industry where a hybrid

between open and closed-loop control has a good likelihood of acceptance, and is applicable to other batch and semi-batch processes as well. This mode of operation is not unlike the mid-course correction scheme proposed in Flores-Cerrillo and MacGregor[22] for emulsion semibatch polymerization, although the methodology for computing the inputs is quite different. An integral component of the advisory system is the multi-tiered optimization strategy, which comprises three hierarchical optimization tiers to handle end-point constraint infeasibilities. A shrinking horizon formulation with an economics-based objective function is used for the optimization tiers. A key bottleneck in implementing real-time advanced control systems is the computational burden associated with solving the large-scale dynamic optimization problems on-line. For obtaining fast solutions to the MHE and the tiered-dynamic optimization problems, a simultaneous dynamic optimization approach is employed, in which the problem is transformed into a sparse nonlinear programming (NLP) problem by discretizing the states and inputs.[23] The MHE-NMPC initialization scheme proposed in Shyamal and Swartz[12] is extended to develop a warm-start strategy for effective initialization of the dynamic optimization problems in the advisory system. The efficacy of the advisory system is evaluated for the EAF process using three case studies, through which four key factors are explored: (1) the impact of re-optimizations at different time points during the batch process operation, (2) handling of potential end-point constraint infeasibilities by increasing the degrees of freedom through extension of the batch duration, (3) the convergence of MHE, and (4) the computational feasibility of a real-time application. The results show a major improvement in the profit obtained when the advisory system is invoked.

The remainder of the article is structured as follows. Sections 2 to 5 describe the components of the advisory system. Specifically, Section 2 describes the first-principles based EAF model used, Section 3 introduces the dynamic optimization problem and model contraction, Section 4 presents the multi-tiered optimization algorithm and its tailoring to the EAF process, and the multi-rate MHE problem along with the arrival cost update formulation is discussed in Section 5. Section 6 describes the integrated advisory system, as well as a multi-tiered

initialization scheme. Section 7 presents the case studies for the electric arc steelmaking process, where the efficacy of the advisory system is demonstrated. Section 8 concludes the paper with a summary of key contributions.

2 Electric arc furnace model

EAF operation is characterized by its complexity and the extreme conditions under which the process operates. A key challenge associated with modeling the EAF process is capturing sufficient detail to adequately track the economics and key states of the process, while yielding fast enough solutions for real-time application. Very detailed computational fluid dynamics (CFD) models are typically available only for a section of the furnace.[24,25] Three-dimensional models are appropriate for engineering and design studies but not suitable for real-time optimal control applications.[26] Models with reasonable simulation times are generally based on simplifying assumptions and typically partition the EAF into multiple zones.[27–29] Detailed physics-based mathematical equations are then employed to describe the heat and mass transfer occurring within and between the zones.

The dynamic first-principles based EAF model used in this paper is based on that proposed by MacRosty and Swartz,[7] which considers four zones, namely, gas, slag-metal, molten-metal and solid scrap zones, as depicted in Figure 1. The *gas zone* represents the gases which fill up the furnace volume above the solid scrap metal. The *slag-metal interaction* zone considers the liquid slag materials and the part of the molten-metal layer which remains in contact with the slag. The *molten-metal* zone considers metals in their molten form once scrap begins to melt. It excludes the part of liquid mass already considered in the *slag-metal* zone. The *solid scrap* zone consists of the scrap mass not yet melted and still in solid form. The process phenomena taking place in the each of the four zones are modeled through the mass and energy balances. The multi-zone system assumes chemical equilibrium in slag-metal

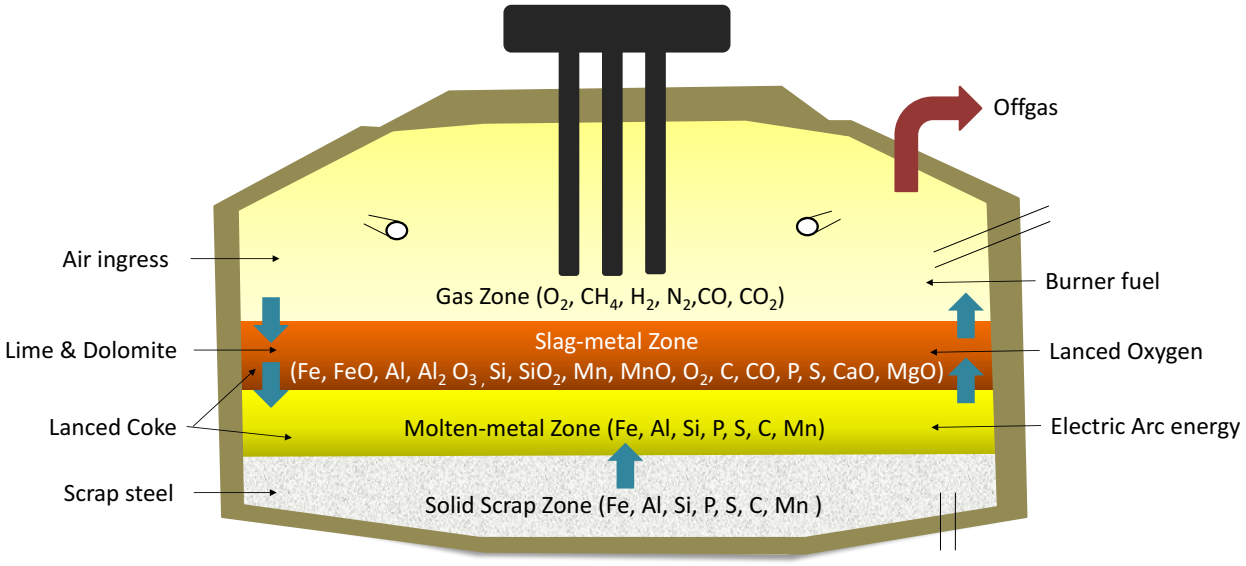

Figure 1: Schematic of the EAF model used in the study showing the 4 zones and associated inputs, outputs and material flows.

and gas zones, with the equilibrium determined by incorporating equations corresponding to the optimality conditions for minimization of the Gibbs free energy. The model also considers slag foaming, which is caused by evolution of carbon monoxide, through empirical relationships. The materials are tracked within and between each zone through mass transfer equations and elemental balances. A key advantage of the equilibrium-based modeling is that one needs to deal with fewer parameters compared to a kinetics-based approach. Energy balances take into account the heat transfer from the arc, chemical reactions, convective and radiative heat transfer. Model parameters were estimated in gPROMS/gEST[30] by employing the maximum likelihood function and using plant data sets. The model comprised 85 differential and 1050 algebraic variables.

A number of modifications to the model were subsequently carried out. Ghobara[20] included models for three oxygen-supplying JetBoxes that were present in a different EAF installation, and also assumed a flat geometry instead of a cone-frustum for scrap melting. The highly nonlinear radiation model which accounted for the radiative heat transfer in the EAF was

removed in Shyamal and Swartz[21] and replaced with a parameter which partitions the arc energy between the molten-metal, scrap metal, furnace roof and walls. Model parameters were re-estimated to match model-generated profiles with the plant data. Also, since the molten-metal zone contains negligible amounts of metal oxides, it was assumed that all the metal oxides are present in the slag-metal zone, and the state variables associated with metal oxides in the molten-metal zone were removed. The modified model contained 29 differential states. The model that was originally formulated and solved in gPROMS[21] was translated to a Python-based CasADi[31] framework for carrying out optimization using the simultaneous approach.[12] Also, a state variable that attained negligible values ($\sim 1.0\times10^{-13}$) was removed and replaced with a parameter. In this work we employ the EAF model used in Shyamal and Swartz[12] in the proposed real-time advisory framework. The DAE model comprises 28 differential and 518 algebraic variables before application of a model contraction procedure described in Section 3. In the next section we describe the general dynamic optimization problem and the proposed multi-tiered optimization strategy to tackle the end-point constraints.

3 Dynamic optimization problem

In this section, we describe the general form of dynamic optimization problem we solve in this work.

3.1 Model formulation

A wide class of first principles-based models can be represented as a differential-algebraic equation (DAE) system, written in semi-explicit form as

$$\dot{\mathbf{x}}(t) = \mathbf{f}'_d(\mathbf{x}(t), \mathbf{z}'(t), \mathbf{u}(t), \mathbf{p}) \tag{1a}$$

$$\mathbf{0} = \mathbf{f}'_a(\mathbf{x}(t), \mathbf{z}'(t), \mathbf{u}(t), \mathbf{p}) \tag{1b}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{z}'(t)) \tag{1c}$$

$$\mathbf{x}(t_i) = \mathbf{x}_{0i} \tag{1d}$$

$$t \in [t_i, t_f] \tag{1e}$$

where $\mathbf{x}$ represents the set of differential state variables and $\mathbf{z}'$ the set of algebraic variables, and $\mathbf{f}'_d$ and $\mathbf{f}'_a$ represent the differential and algebraic functions respectively of the DAE process model. The inputs for the system are represented by $\mathbf{u}$, and the system parameters are denoted by $\mathbf{p}$. The equations $\mathbf{h}$ describe the mapping of the differential and algebraic states to system outputs $\mathbf{y}$. The system evolves from given initial time t_i to final time t_f for given initial states $\mathbf{x}_{0i}$.

In direct optimization approaches,[23] the inputs are parameterized, with piecewise constant or piecewise linear function being common choices. The high dimension of DAEs arising in the modeling of large-scale process systems has prompted the development of model reduction techniques.[32,33] Here, we describe model contraction which preserves the integrity of the original DAE system, but reduces the search space through elimination of variables through the equality constraints, and is a step that is available in solver platforms such as CasADi.

3.2 Model contraction

We describe as model contraction the procedure of eliminating a subset of algebraic variables by converting them into dependent variables $\mathbf{d}$. Such an elimination process will also remove an equal number of algebraic equations. Symbolic preprocessing schemes for carrying out this procedure are described in Magnusson.[34] The contracted model is given as

$$\dot{\mathbf{x}}(t) = \mathbf{f}_d(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \tag{2a}$$

$$\mathbf{0} = \mathbf{f}_a(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \tag{2b}$$

$$\mathbf{d}(t) = \mathbf{f}_e(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \tag{2c}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{d}(t)) \tag{2d}$$

$$\mathbf{x}(t_i) = \mathbf{x}_{0i} \tag{2e}$$

$$t \in [t_i, t_f], \tag{2f}$$

where, $\mathbf{d}$ is a subset of $\mathbf{z}'$, and $\mathbf{z}$ are the remaining algebraic variables. It is to be noted that the above model contraction preserves the model fidelity; however, the model sparsity is sacrificed to obtain a more compact DAE model than the original one. A useful outcome of this procedure is that eq 2c describing the evolution of $\mathbf{d}$ can be considered external to the model with $\mathbf{d}$ post-calculated. Although we represent the new differential equation function as $\mathbf{f}_d$, the dimension of $\mathbf{f}_d$ is equal to dimension of $\mathbf{f}'_d$. If $\mathbf{d}$ appears in constraints or the objective function, then the explicit expression $\mathbf{f}_e$ is used instead. Such a substitution is potentially needed when formulating a dynamic optimization or an MHE problem.

We have implemented model contraction using the CasADi framework.[31] The reformulation is achieved by applying the in-built function 'eliminate_alg()' to the DAE system (described using the in-built 'DaeBuilder class' of CasADi). The function symbolically reformulates the DAE system and directly provides the contracted DAE system, with the elimination process

transparent to the user.

Remark: Model contraction is also implemented in the software package, JModelica,[35] and gPROMS,[30] implements a similar internal restructuring referred to as 'model pruning' for obtaining a minimum-sized model for simulation and optimization tasks. However, for implementation of model contraction from the ground up, the user guides of CasADi and JModelica (and the references given there) are useful resources.

3.3 Optimization formulation

Given the current states $\mathbf{x}_i$ provided by a state estimation strategy at time t_i, the dynamic optimization problem takes the following general form

$$\max_{\mathbf{u}(t),\, t\in[t_i,t_f]} \quad \Phi(t_f) := \int_{t_i}^{t_f} \psi(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p})dt \tag{3a}$$

$$\text{subject to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}_d(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}), \tag{3b}$$

$$\mathbf{0} = \mathbf{f}_a(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}), \tag{3c}$$

$$\mathbf{u}^L \leq \mathbf{u}(t) \leq \mathbf{u}^U, \tag{3d}$$

$$\mathbf{x}^L \leq \mathbf{x}(t) \leq \mathbf{x}^U, \quad \forall t \in [t_i, t_f] \tag{3e}$$

$$\mathbf{x}_f^L \leq \mathbf{x}(t_f) \leq \mathbf{x}_f^U, \tag{3f}$$

$$\mathbf{z}^L \leq \mathbf{z}(t) \leq \mathbf{z}^U, \tag{3g}$$

$$\mathbf{x}(t_i) = \mathbf{x}_i, \tag{3h}$$

where Φ represents the objective function. We aim to find the optimal inputs $\mathbf{u}$ so as to maximize Φ while satisfying the model equations (3b and 3c), and constraints on inputs (eq 3d), differential state variables (eqs 3e and 3f) and algebraic variables (eq 3g). The superscripts L and U represent the lower and upper bounds respectively. This general representation

admits various types of objective function such as an economics-based objective function, set-point tracking MPC objective function, and so forth.

A key bottleneck in solving the above dynamic optimization problem in batch control applications is the end-point constraint eq 3f. This is typically hard to satisfy since it is possible that with a given $\mathbf{x}_i$, no set of inputs can steer the states to the region $[\mathbf{x}_f^L, \mathbf{x}_f^U]$ within a fixed time horizon $[t_i, t_f]$. An immediate consequence can be optimization solvers flagging an infeasible solution, leading to an overall halt of an advanced control procedure. This is highly undesirable from a practical implementation standpoint, and an effective strategy is needed to generate useful optimization results even when faced with such situations. One way to tackle the problem is to include the final time t_f in the decision variable space for the optimization problem, which for a fixed number of control variables stretches/contracts control intervals to find the optimal t_f.[36] Apart from an increase in the problem complexity, a couple of key issues arise. First, accuracy of numerical discretization will decrease if control intervals are stretched. Although an adaptive discretization scheme can be employed, the problem complexity will increase substantially. Second, there will be a mismatch in the sampling times and control intervals, which will pose implementation challenges. Thus, development of a resilient real-time strategy is necessary when simply removing the end-point constraint is not acceptable. Having described the general dynamic optimization problem, in the next subsection we propose a multi-tiered optimization strategy for dynamic optimization with end-point constraints.

4 Multi-tiered optimization

In this section, we present a strategy for handling a dynamic optimization problem with end-point constraints through multi-tiered optimization. The strategy incorporates three sequential dynamic optimization tiers where the second and third tier come into effect when

the previous tier fails to give a solution. It is particularly useful for end-point constraints for which the likelihood of feasibility increases with batch time, such as residual solid scrap or carbon content in the molten steel in electric arc steelmaking. An important characteristics of the strategy is that for problems in which the end-point constraints limit problem feasibility, application of the tiered optimization approach terminates with a feasible solution to a modified problem, that is generally also acceptable from an industrial application perspective. We describe each of the three tiers in the subsequent subsections.

4.1 Tier 1: Direct optimization

In this tier, we directly attempt to solve the original dynamic optimization problem given in eqs 3a-3h. Any of the three dynamic optimization approaches, viz. sequential, simultaneous or multiple-shooting, can be employed to solve the problem. However, the NLP solvers used in the three approaches are allowed to carry out the optimization up to a maximum number of iterations max_{iter} in order for the optimization calculation to terminate within a reasonable time. The max_{iter} is a tunable parameter and a suitable value can be chosen depending on the problem size and computational time needed for each solver iteration. If the solver declares optimality upon termination, then we do not move to the next tier. However, if the solver terminates due to infeasibility or the maximum number of iterations being reached, then we proceed to Tier 2.

4.2 Tier 2: Feasibility through horizon extension

In Tier 2, the original dynamic optimization problem of Tier 1 is modified to enhance the likelihood of achieving feasibility by extending the time horizon $t \in [t_i, t_f]$ in integral steps.

The problem with an extension of N_e integer time steps is given as

$$\max_{\mathbf{u}(t),\, t\in[t_i, t_f+N_e\Delta T]} \quad \Phi(t_f + N_e\Delta T) := \int_{t_i}^{t_f+N_e\Delta T} \psi(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p})dt \tag{4a}$$

$$\text{subject to} \quad \text{Extended set of equations :}$$

$$\text{eqs 3b-3d, 3g, 3h,} \tag{4b}$$

$$\mathbf{x}^L \leq \mathbf{x}(t) \leq \mathbf{x}^U, \quad \forall t \in [t_i, t_f + N_e\Delta T) \tag{4c}$$

$$\mathbf{x}_f^L \leq \mathbf{x}(t) \leq \mathbf{x}_f^U, \quad t = t_f + N_e\Delta T. \tag{4d}$$

This extension provides additional inputs for manipulation by the dynamic optimization. These additional degrees of freedom provide a straightforward way to enhance the achievement of feasibility for systems where end-point constraints are more likely to be satisfied when batch length is extended. We describe a serial and a parallel implementation for Tier 2:

- **Serial implementation**: First, a 1 time step extension problem ($N_e = 1$) is formulated and solved. If it remains infeasible or the maximum number of iterations is reached, then a second problem where $N_e = 2$ is solved. This sequence of problems is continued up to a specified limited number of time step extensions N_{ext}. If a feasible solution is not achieved within the iteration limit at the maximum extended batch length, we proceed to Tier 3; otherwise we stop in Tier 2 and report the solution obtained with the associated extension (number of time steps: $N_{e_{min}}$ value) required to achieve feasibility.

- **Parallel implementation**: We formulate and solve N_{ext} extended problems in parallel. This effectively uses available parallel computational architecture. Solver exit messages for all the N_{ext} solves are collected, and if none of them is solved to full optimality, we proceed to Tier 3. If one or more problems are solved, then we choose the extension $N_{e_{min}}$ associated with the solve with the minimum extension value.

Both the implementations attempt to find the minimum integral extension required to achieve feasibility. It is to be noted that the two implementations are attempting to solve a mixed-integer dynamic optimization (MIDO)[37] problem via explicit enumeration. If feasibility in not achieved, then we move to the final Tier 3 which is described below.

4.3 Tier 3: End-point constraint relaxation

In this tier, we reformulate the extended problem of Tier 2 by softening[38] the end-point constraint at $t_f + N_{ext}\Delta T$ using slack variable $\boldsymbol{\epsilon}$. The reformulated problem is given as

$$\max_{\mathbf{u}(t),\, t\in[t_i, t_f+N_{ext}\Delta T]} \quad \Phi(t_f + N_{ext}\Delta T) := \int_{t_i}^{t_f+N_{ext}\Delta T} \psi(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p})dt + ||\boldsymbol{\epsilon}||^2_{D^{-1}} \tag{5a}$$

$$\text{subject to} \quad \text{eqs 4b and 4c,} \tag{5b}$$

Softened end-point constraint:

$$\mathbf{x}^L_{t_f+N_{ext}\Delta T} - \boldsymbol{\epsilon} \leq \mathbf{x}(t_f + N_{ext}\Delta T) \leq \mathbf{x}^U_{t_f+N_{ext}\Delta T} + \boldsymbol{\epsilon} \tag{5c}$$

$$\boldsymbol{\epsilon} \geq \mathbf{0}, \tag{5d}$$

where the modified objective function contains the additional least squares penalty term for the slack variables. The weighting for each of the slack variables in the penalty term is given by diagonal matrix D. The soft constraint eq 5c accommodates any infeasibility caused due to end-point constraint violation at $t_f + N_{ext}\Delta T$.

This relaxed problem will be feasible if feasibility is limited by end-point constraints, and we stop at Tier 3. The use of three tiers is designed to arrive at a reasonable solution to the original problem in circumstances where the original problem is not feasible; with the solutions thus obtained implemented on the plant. The working of the three tiers is summarized in Figure 2. A multi-tiered optimization approach for calculating set-points

in a supervisory system with prioritized objectives was proposed in Swartz,[39] and tiered approaches were also employed by Chong and Swartz[40,41] to handle multiple objectives in frameworks for optimal operation under partial plant shutdown. The optimization tiers in these studies were used to handle prioritized objectives with consideration of feasibility. The present tiered algorithm is used to resolve end-point constraint infeasibility through batch time extension and end-point constraint relaxation, with the former considered as more desirable (up to a maximum allowable extension).

A potential alternative to tiered-optimization is a single optimization with suitable weights (penalties) for different objectives. An advantage of our proposed approach is that it captures the logic of prioritization precisely rather than through a penalty-based approach where priorities would be compromised to some degree. Additionally, in the case of a single optimization, the values of the weights become tuning parameters. In the next subsection we describe the specific application of the multi-tiered optimization to EAF operation.

4.4 EAF implementation

We aim to maximize the economics of the EAF batch process by solving a dynamic optimization problem based on a first-principles EAF model. The optimization is tackled through the multi-tiered optimization strategy, with the result provided to the operator. The first tier, *direct optimization*, solves a shrinking horizon dynamic optimization problem with an economic objective and a constraint on the end point solid scrap mass m_{ss}. The second and third tiers come into effect if infeasibility is detected in Tier 1. The second tier attempts to achieve feasibility through horizon extension, whereas the third tier solves a relaxed problem.

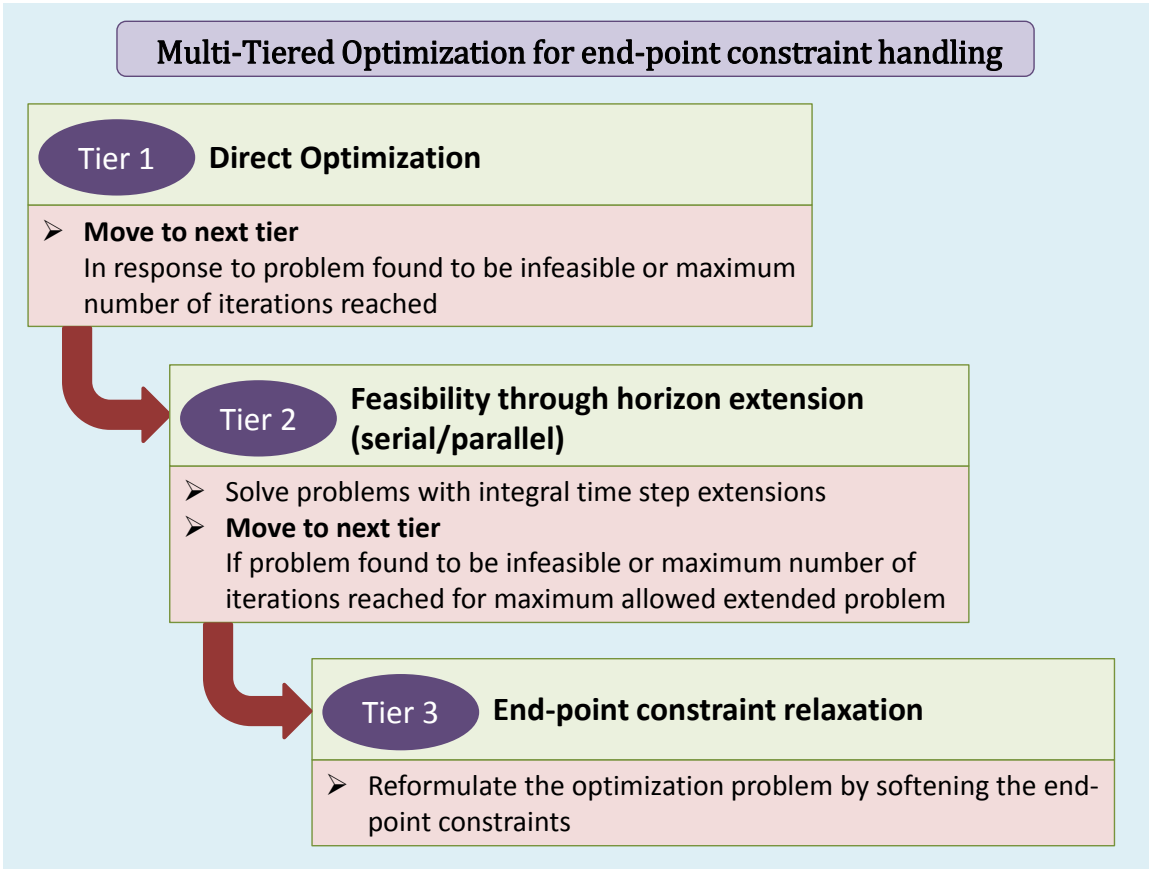

Figure 2: Multi-tiered optimization strategy.

4.4.1 Tier 1: Direct optimization

In the first tier, we aim to maximize economics subject to the EAF model (2) in order to find the optimal input trajectories. The objective function for the EAF is to maximize profit at the end of the EAF batch, which is given as

$$\begin{aligned}\Phi(t_f) := c_0 M_{steel}(t_f) - \Bigg(& c_1 \int_{t_i}^{t_f} P dt + c_2 \int_{t_i}^{t_f} F_{CH_4,brnr} dt + c_3 \int_{t_i}^{t_f} F_{C_{lance}} dt \\ & + c_4 \int_{t_i}^{t_f} F_{C_{charge}} dt + \ c_5 \int_{t_i}^{t_f} (F_{O_2,Jetbox1} + F_{O_2,Jetbox2} + F_{O_2,Jetbox3}) dt \\ & + c_6 \int_{t_i}^{t_f} F_{CaO} dt + \ c_7 \int_{t_i}^{t_f} F_{Dolomite} dt + \ c_8 \int_{t_i}^{t_f} F_{1stCharge} dt + \ c_9 \int_{t_i}^{t_f} F_{2ndCharge} dt \Bigg), \end{aligned} \quad (6)$$

where M_{steel} is the amount of molten steel, with its corresponding selling price denoted by c_0. M_{steel} produced at t_f is the major source of revenue and profit is calculated by subtracting from it the total cost incurred for all the inputs applied throughout the duration of the batch. The ten utilized inputs are: electric arc power P, flow of burner natural gas $F_{CH_4,brnr}$, lanced carbon $F_{C_{lance}}$, charged carbon $F_{C_{charge}}$, flow of oxygen from three JetBoxes $(F_{O_2,Jetbox1}, F_{O_2,Jetbox2}, F_{O_2,Jetbox3})$, lime F_{CaO}, dolomite $F_{Dolomite}$ and second scrap charge $F_{2ndCharge}$. We discretize the integral in eq 6 with no loss in accuracy due to piecewise constant input trajectories.

The input constraints are given as

$$P^{min}(t) \leq P(t) \leq P^{max}(t) \quad (7a)$$

$$F_j^{min}(t) \leq F_j \leq F_j^{max}(t), \quad (7b)$$

where superscripts min and max denotes the lower and upper limits respectively. The input constraint have been implemented to ensure that we obtain the optimal input sequence in

the realistic operating range of the furnace. Path constraints

$$T_{wall} \leq T^{max}, \tag{8}$$

$$T_{roof} \leq T^{max}, \tag{9}$$

ensure that the furnace wall (T_{wall}) and roof temperatures (T_{roof}) are below the maximum allowed temperature T^{max}. Additionally, we have end-point constraints

$$m_{ss}(t_f) \leq \delta_{ss}, \tag{10}$$

$$y_{carbon}(t_f) \leq Y_c^{max}, \tag{11}$$

for the optimization problem. Eq 10 ensures that all the scrap left is essentially melted by the end of the batch, where $m_{ss}(t_f)$ represent the amount of solid scrap left at final time t_f, and δ_{ss} is the specified tolerance. Eq 11 ensures that the carbon content of the steel $y_{carbon}(t_f)$ at t_f is within the specified level Y_c^{max}. The second and third tiers are applied with respect to the limiting end-point constraint eq 10, as described in sections 4.2 and 4.3.

5 Multi-rate Moving Horizon Estimation

For carrying out an operator-triggered multi-tiered optimization at any time step t_i, it is important to reconstruct the states $\mathbf{x}_i$ using the available measurements up to the current time. MHE reconstructs the state by solving a dynamic optimization problem subject to the model and other specified constraints.[16] Since MHE solves a dynamic optimization problem to determine the state estimates, efficient nonlinear programming (NLP) solvers can be used to arrive at the solutions quickly. MHE uses a past fixed-sized moving window of measurements to keep the problem size reasonable. However, when problem size is not a major concern, then an expanding horizon least-squares estimation can be used to avoid loss

of information due to a moving horizon.[42] The respective problem constitutes full information least-squares estimation where all measurements up to the current time are considered in the formulation.[16] It is to be noted that the use of a past history of measurements by MHE provides a straightforward way to use measurements with different sampling rates.[21,43–45] The use of all the available measurements potentially leads to an increase in the system observability and also a reduction in estimation errors.[44,45] The next subsections describe the model used in the MHE formulation, the least-squares problem and the arrival cost calculation.

5.1 Model for state estimation

We consider the process formulated as a stochastic DAE system

$$\dot{\mathbf{x}}(t) = \mathbf{f}_d(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) + \mathbf{w}_t(t) \tag{12a}$$

$$\mathbf{0} = \mathbf{f}_a(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \tag{12b}$$

$$\mathbf{d}(t) = \mathbf{f}_e(\mathbf{x}(t), \mathbf{z}(t), \mathbf{u}(t), \mathbf{p}) \tag{12c}$$

$$\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t), \mathbf{z}(t), \mathbf{d}(t)) + \mathbf{v}_t(t), \tag{12d}$$

where $\mathbf{w}_t$ represents the errors in the differential equations constituting process noise, and is useful for accounting for disturbances and model mismatch. Eq 12d denotes the measurement equation where $\mathbf{v}_t$ accounts for measurement noise. Typically both $\mathbf{w}_t$ and $\mathbf{v}_t$ are modeled as a normally distributed uncorrelated sequence of random variables.

Discretization of the above DAE system is typically carried out for optimization computations. Since discretization is performed according to the optimization strategy employed (simultaneous, multiple shooting), we consider here the conceptual discretization of the DAE system in stating the constrained least-squares MHE formulation.[46] We also assume that in-

puts $\mathbf{u}$ are only manipulated at discrete time points k. The subscript k is introduced here to distinguish variable values at discrete time points from continuous values. Thus, using eqs 12a-12b, and removing $\mathbf{p}$ for ease of representation, we obtain

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \int_{t_k}^{t_{k+1}} \mathbf{f}_d(\mathbf{x}(t), \mathbf{u}_k)dt + \int_{t_k}^{t_{k+1}} \mathbf{w}_t(t)dt, \tag{13}$$

which is represented as

$$\mathbf{x}_{k+1} = \mathbf{F}_d(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k. \tag{14}$$

We also obtain the discrete time measurement equation as

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k. \tag{15}$$

We note here that eq 14 represents a closed form solution of the DAE system which is not always possible to obtain. Here it is viewed conceptually, with the algebraic and dependent variables/equations considered to be eliminated. Here, $t_k = k\Delta T$, where ΔT is the sampling period.

5.2 Constrained least-Squares formulation

We consider the process at a given time instant t_i with a past history of measurements and inputs available up to t_i. An MHE window of fixed N sampling times is used. Here, we write the MHE formulation for a multi-rate measurement structure. The key assumptions are that the time points of slow (infrequent) measurements coincide with those of the fast (frequent) ones, and there are no delays associated with availability of measurements. The vector of fast measurements is denoted as $\mathbf{y}_k^F$, while $\mathbf{y}_k^{SF}$ represents a measurement vector containing both the slow and fast measurements. Thus, a measurement structure for MHE can be

represented, for example, as $\{\mathbf{y}_{i-N}^{SF}, \mathbf{y}_{i-N+1}^{F}, \mathbf{y}_{i-N+2}^{SF}, \ldots, \mathbf{y}_{i-1}^{F}, \mathbf{y}_{i}^{SF}\}$. In the example, at time instants $\{t_{i-N}, t_{i-N+2}, t_i\}$ both the slow and fast measurements are available, whereas only fast measurements are available at time instants $\{t_{i-N+1}, t_{i-1}\}$. The measurement structure of the example can be easily modified to accommodate different multi-rate sampling sequences. As we proceed in time, new measurements are added to the structure, and the earliest ones dropped.

The multi-rate MHE problem for a discrete-time system is given as

$$\min_{\mathbf{x}_{i-N}, \mathbf{w}_k} \sum_{k=i-N}^{i-1} ||\mathbf{w}_k||^2_{Q^{-1}} + \sum_{\substack{k=i-N \\ k\in\mathbb{I}_F}}^{i} ||\mathbf{v}_k^F||^2_{(R^F)^{-1}} + \sum_{\substack{k=i-N \\ k\in\mathbb{I}_{SF}}}^{i} ||\mathbf{v}_k^{SF}||^2_{(R^{SF})^{-1}} + ||\mathbf{x}_{i-N} - \hat{\mathbf{x}}_{i-N}||^2_{S_i^{-1}} \tag{16}$$

$$\text{s.t. } \mathbf{x}_{k+1} = \mathbf{F}_d(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{w}_k, \quad k = i-N, \ldots, i-1 \tag{17a}$$

$$\mathbf{y}_k^F = \mathbf{h}_k^F(\mathbf{x}_k) + \mathbf{v}_k^F, \qquad k \in \mathbb{I}_F \tag{17b}$$

$$\mathbf{y}_k^{SF} = \mathbf{h}_k^{SF}(\mathbf{x}_k) + \mathbf{v}_k^{SF}, \quad k \in \mathbb{I}_{SF} \tag{17c}$$

$$\mathbf{x}^{LB} \leq \mathbf{x_k} \leq \mathbf{x}^{UB}, \tag{17d}$$

$$\mathbf{w}^{LB} \leq \mathbf{w_k} \leq \mathbf{w}^{UB}, \tag{17e}$$

where $\mathbb{I}_F$ and $\mathbb{I}_{SF}$ represent the sets of sampling times where only fast measurements (super script F) and both slow and fast measurements (superscript SF) are available respectively. $\hat{\mathbf{x}}_{i-N}$ denotes the *a priori* estimate of the state variable value at the earliest time point in the moving horizon. We represent two separate measurements functions $\mathbf{h}_k^F$ and $\mathbf{h}_k^{SF}$ to map the states $\mathbf{x}_k$ to the sensor measurements $\mathbf{y}_k^F$ and $\mathbf{y}_k^{SF}$ respectively. Similarly, $\mathbf{v}_k^F$ and $\mathbf{v}_k^{SF}$ represent the associated measurement noise. Eqs 17d and 17e provide the lower $[\mathbf{x}^{LB}, \mathbf{w}^{LB}]$

and upper bounds $[\mathbf{x}^{UB}, \mathbf{w}^{UB}]$ on the state variables $\mathbf{x}$ and model noise $\mathbf{w}$ respectively. The covariance matrices associated with the model noise, measurement noise (for F and SF) and initial state error are denoted here as Q, R^F, R^{SF} and S_i respectively. The state estimates $\mathbf{x}_i$ at t_i are obtained by solving the least squares problem.

It can be observed that the MHE least squares objective function is composed of four terms, where the first three are used to minimize weighted model and measurement errors. The fourth term, the arrival cost, minimizes the initial state discrepancy. The arrival cost term is important because we are dropping measurements due to the moving nature of the MHE. The initial state error covariance is also generally updated every time step, thus the subscript i is used for the S matrix. However, the update is not entirely necessary when the system is strongly observable.[47] Zavala[48] showed that a single parameter penalizing the arrival cost is sufficient to achieve convergence. In the next subsection, we describe the EKF covariance update formula[16] which is commonly used for updating the initial state covariance matrix.

5.3 Arrival cost update

To update the arrival cost covariance matrix S using the EKF update formula, we need a discrete time model linearized at every sampling time. We first consider the continuous time stochastic DAE model eq 12. Linearizing eq 12 around a given time t_k gives (suppressing $\mathbf{p}$ for ease of readability)

$$\dot{\mathbf{x}}(t) = P_k\mathbf{x}(t) + J_k\mathbf{z}(t) + K_k\mathbf{u}(t) + \mathbf{w}_t(t) \tag{18a}$$

$$\mathbf{0} = G_k\mathbf{x}(t) + H_k\mathbf{z}(t) + L_k\mathbf{u}(t), \tag{18b}$$

$$\mathbf{d}(t) = O_k\mathbf{x}(t) + T_k\mathbf{z}(t) + V_k\mathbf{u}(t), \tag{18c}$$

$$\mathbf{y}(t) = E_k\mathbf{x}(t) + F_k\mathbf{z}(t) + M_k\mathbf{d}(t) + \mathbf{v}_t(t). \tag{18d}$$

where matrices $P_k, J_k, K_k, G_k, H_k, L_x, O_k, T_k, V_k, E_k, F_k$ and M_k are obtained by computing respective Jacobian values at the linearized point. Although, a discretization can now be carried out by an appropriate integration of the continuous system through a zero-order hold on $\mathbf{u}(t)$, this transformation involves computing the matrix exponential. For large-scale systems where the system matrices are prone to be ill-conditioned, we employ an alternate way of transformation which uses the explicit Euler scheme for discretization. Although the approach is less accurate, we can completely avoid the complications associated with calculating matrix exponentials. Using the explicit Euler scheme, we obtain the discretized form as

$$\frac{\mathbf{x}_{k+1} - \mathbf{x}_k}{\delta_1} = P_k\mathbf{x}_k + J_k\mathbf{z}_k + K_k\mathbf{u}_k + \mathbf{w}_k, \tag{19a}$$

$$\mathbf{0} = G_k\mathbf{x}_k + H_k\mathbf{z}_k + L_k\mathbf{u}_k, \tag{19b}$$

$$\mathbf{d}_k = O_k\mathbf{x}_k + T_k\mathbf{z}_k + V_k\mathbf{u}_k, \tag{19c}$$

$$\mathbf{y}_k = E_k\mathbf{x}_k + F_k\mathbf{z}_k + M_k\mathbf{d}_k + \mathbf{v}_k, \tag{19d}$$

where δ_1 denotes the Euler step size. Using eq 19b to eliminate $\mathbf{z}_k$ from eq 19a, eq 19c and eq 19d, and further using eq 19c to substitute $\mathbf{d}_k$ in eq 19d and rearranging terms, gives the discretized state space model to be used for EKF updates,

$$\mathbf{x}_{k+1} = \delta_1(P_k - J_k{H_k}^{-1}G_k + I)\mathbf{x}_k + \delta_1(K_k - J_k{H_k}^{-1}L_k)\mathbf{u}_k + \mathbf{w}_k \tag{20a}$$

$$\begin{aligned}\mathbf{y}_k = (E_k - F_k{H_k}^{-1}G_k - M_kT_k{H_k}^{-1}G_k + M_kO_k)\mathbf{x}_k \\ + (M_kV_k - F_k{H_k}^{-1}L_k - M_kT_k{H_k}^{-1}L_k)\mathbf{u}_k + \mathbf{v}_k\end{aligned}$$

where I is an identity matrix of dimension equal to number of states. Replacing $[\delta_1(P_k - J_k{H_k}^{-1}G_k + I)]$, $[\delta_1(K_k - J_k{H_k}^{-1}L_k)]$, $[(E_k - F_k{H_k}^{-1}G_k - M_kT_k{H_k}^{-1}G_k + M_kO_k)]$ and $[(M_kV_k - F_k{H_k}^{-1}L_k - M_kT_k{H_k}^{-1}L_k)]$ with A_k, B_k, C_k and D_k respectively, yields a time-

varying discrete state space model in standard form as

$$\begin{aligned}\mathbf{x}_{k+1} &= A_k\mathbf{x}_k + B_k\mathbf{u}_k + \mathbf{w}_k \\ \mathbf{y}_k &= C_k\mathbf{x}_k + D_k\mathbf{u}_k + \mathbf{v}_k.\end{aligned} \tag{21}$$

For the above system, an EKF covariance propagation equation is given as,[16]

$$S_{k+1} = Q + A_k[S_k - S_kC_k^T(R + C_kS_kC_k^T)^{-1}C_kS_k]A_k^{-1}. \tag{22}$$

Although, the superscripts 'SF' and 'F' have been suppressed in eq 22 for R and C, and in eq 21, appropriate dimensions according to the multi-rate measurement structure need to be considered. It is to be noted that other filters such as the constrained particle filter[49] and unscented KF[50] can also be employed to update S_i. The update is not carried for batch process with sampling times t_i ($i \leq N$), since measurements are not yet dropped.

5.4 Disturbance handling

To tackle unmeasured disturbances with nonzero mean and plant-model mismatch, the model states are augmented with disturbance states $\mathbf{x}_k^d$ to construct a modified system for MHE

$$\mathbf{x}_{k+1} = \mathbf{F}_d(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{B}_d\mathbf{x}_k^d + \mathbf{w}_k, \tag{23a}$$

$$\mathbf{y}_k^F = \mathbf{h}_k^F(\mathbf{x}_k) + \mathbf{v}_k^F, \tag{23b}$$

$$\mathbf{y}_k^{SF} = \mathbf{h}_k^{SF}(\mathbf{x}_k) + \mathbf{v}_k^{SF}, \tag{23c}$$

$$\mathbf{x}_{k+1}^d = \mathbf{x}_k^d + \mathbf{w}_{dk}, \tag{23d}$$

where $\mathbf{x}_k^d$ is driven by a new set of white noise variables $\mathbf{w}_{dk} \sim \mathcal{N}(0, \mathbf{Q}_d)$. A well chosen $\mathbf{B}_d$ ensures that the corresponding state equations have the right impact due to the movement of

the disturbance states $\mathbf{x}_k^d$. More detail on the disturbance model and its effect are discussed in Ji and Rawlings.[45]

6 Real-time advisory system

In this section, we describe how the components of the advisory system described in Sections 2-5 are assembled to form an integrated real-time decision-support system. We also describe an initialization strategy for the MHE and economic dynamic optimization problems, as well as the software implementation followed in this work.

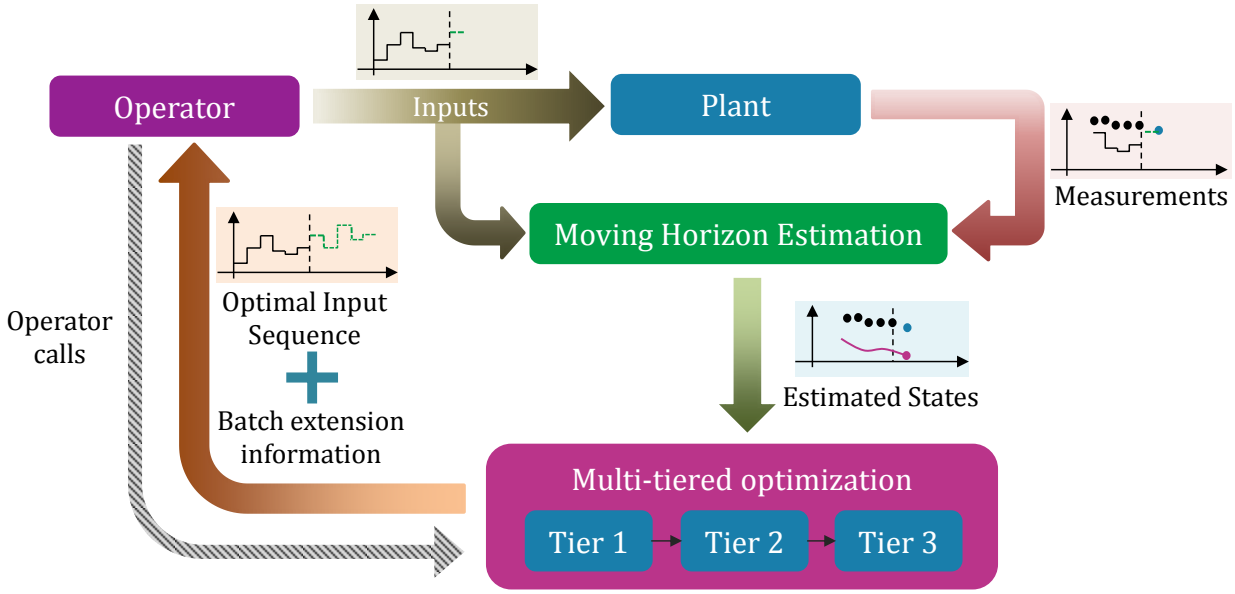

Figure 3: Real-time advisory system.

6.1 Integrated decision-support system

The integrated real-time advisory system is illustrated in Figure 3. The main objective of the real-time advisory system is to provide an input sequence as decision support to the EAF

operators when needed. A key component of the advisory system is the multi-rate MHE which runs in parallel with the EAF to obtain the state estimates at every sample time. The multi-tiered optimizer is called upon only when the operator triggers the advisory system. The advisory system uses a multi-tiered initialization scheme (described in Section 6.2), in combination with state-of-the-art NLP solvers, to obtain fast solutions to the dynamic optimization problems. The operator may choose to implement the input recommendation provided by the advisory system up to a certain time in the batch by implementing only the initial subset of the suggested input moves, and then trigger the advisory system again at a later time. The operator may alternatively choose not to implement the calculated inputs, or to deviate from the suggested input sequence. However, the actual input sequence applied to the plant, regardless of the operator's decision, is used by the MHE (applied at every sample time) so that the state estimates are consistent with the actual process operation. The tiered optimization scheme also provides information related to lengthening of batch if required, so that operators are aware of the predicted impact and can take remedial action at an early stage.

6.2 Multi-tiered initialization strategy

For decision support to be available to the operators in real-time, solving the optimization problems of the tiers and MHE problem quickly is very important. Since successive MHE and optimization problems have similar structures, it is useful to use the past solves effectively to generate good warm start points for the current solves. Warm-start strategies have been successfully employed by multiple researchers to solve MHE and NMPC problem on-line. For obtaining quick optimal solutions to online MHE-NMPC problems, various initialization strategies such as initialization based on parametric sensitivities and shift initialization can be employed.[51] Zavala and Biegler introduced an advanced-step NMPC and fast MHE strategy for getting quick approximate solutions to the online optimal control problems based on

NLP sensitivity.[52,53] Real-time algorithms have been proposed where solution of a problem obtained at the previous sampling time is shifted, and used to construct and carry out a single iteration of the controller optimization.[54,55] A comprehensive review on the real-time algorithms for solving MHE and NMPC problems is presented in Diehl et al.[51] In a recent work, Shyamal and Swartz[12] proposed a novel initialization scheme to solve on-line MHE-NMPC problems quickly to optimality by effectively using the time between sample times. Here, we adapt the scheme for our advisory system application. We solve predicted optimization problems in the background and then use the solutions for warm-starting the actual problems. The proposed multi-tiered initialization strategy generates warm-start points for both the primal and dual variables.

We consider the current time instant (on-line time) as t_k where the state estimates $\mathbf{x}_k$ and the inputs applied to the plant $\mathbf{u}_k$ are available. The time available to us before the next sampling time t_{k+1} is referred to here as background time. The superscripts for slow and fast measurements are suppressed here for ease of readability. Now, we describe the multi-tiered initialization strategy for the real-time advisory system:

In background, between t_k and t_{k+1}:

- **Tier 1**: While keeping $\mathbf{w}_k$ as $\mathbf{0}$, carry out a disturbance-free forward model simulation $\bar{\mathbf{x}}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k)$ using $\mathbf{x}_k$ and $\mathbf{u}_k$. Obtain the predicted measurement $\bar{\mathbf{y}}_{k+1} = \mathbf{h}(\bar{\mathbf{x}}_{k+1})$ and pass it to Tier 2.

- **Tier 2**: Use $\bar{\mathbf{y}}_{k+1}$ to construct a predicted MHE problem. Warm start the predicted MHE problem by using the solution obtained from the last actual MHE solve, removing the primal and dual value associated with the first control stage, and then adding the guesses for the primal variables values for the new control stage using a forward simulation. Solve the predicted MHE problem and obtain the predicted state estimates $\tilde{\mathbf{x}}_{k+1}$ to be passed to Tier 3. Also, store the optimal primal and dual values $\tilde{s}_k^{mhe}$.

- **Tier 3**: Use $\tilde{\mathbf{x}}_{k+1}$ to construct the predicted optimization problems in the multi-tiered optimization strategy. Initialize the predicted problem by taking the primal and dual variable values from the last actual solve and dropping the variables values associated with the first control interval. Solve the optimization problem and store the solutions obtained from optimization tiers 1 to 3 as $\tilde{s}_k^{opt1}$, $\tilde{s}_k^{opt2}$, and $\tilde{s}_k^{opt3}$ (both primal and dual).

On-line, at t_{k+1}:

- **Tier 1**: Obtain the current plant measurements $\mathbf{y}_{k+1}$ and construct the actual MHE problem. Use $\tilde{s}_k^{mhe}$ to warm start the problem; solve it to get the actual current state estimates $\mathbf{x}_{k+1}$ and pass it to Tier 2.

  If operator triggers the advisory system, proceed to Tier 2; otherwise return to the background step.

- **Tier 2**: Construct and solve optimization problems associated with the multi-tiered optimization strategy by using $\tilde{s}_k^{opt1}$, $\tilde{s}_k^{opt2}$, and $\tilde{s}_k^{opt3}$ for initialization. Obtain the optimal input sequence $\{\mathbf{u}_{k+1}, \ldots, \mathbf{u}_{t_f}\}$ and send it to the operator. Return to the background step.

While the effectiveness of the multi-tiered strategy would in general depend on how close the predicted and actual problems are, we obtained substantial reductions in solve times with this approach for all the case studies considered. Also, the strategy can effectively be implemented for any of the three dynamic optimization approaches viz. single shooting, multiple shooting and simultaneous (details of the approaches in Biegler[23]).

An important aspect of the proposed advisory system algorithm lies in providing the freedom of choosing trigger points to the operator rather than this being prescribed by the algorithm itself, which is well-suited for electric arc steelmaking. However, a recommendation for optimization trigger points based on the state estimates and covariance could be implemented

through a process monitoring layer that could detect the quality of the model and states as the batch progresses. If state bound violations are predicted, or the state of the system indicates that the model is no longer representative of the plant, an optimization trigger can be indicated to the operator. A strategy of this type would be a useful avenue for further research.

6.3 Implementation

Model contraction is first used in CasADi to remove 397 algebraic variables from the original DAE model containing 28 differential and 518 algebraic variables. The eliminated algebraic variables become the dependent variables defined by explicit functions of only the differential state and remaining algebraic variables. The advisory system is implemented for a discretized model using the Python front end of CasADi. We use the implicit Euler scheme in a simultaneous solution (direct transcription) approach with 7 time steps per minute (the sampling time for the EAF system is 1 minute) for discretization. The infinite dimensional optimization problem is thus converted to a sparse large-scale NLP problem. We employ the interior-point solver IPOPT[56] (with MA27 as the linear solver) to solve the NLP. We have also used IDAS (part of SUNDIALS suite of solvers[57]) for carrying out forward simulations for generating measurements.

7 Case studies

In this section, three case studies are presented to demonstrate potential benefits of the proposed advisory system. The studies correspond to scenarios to evaluate the effect of advisory system executions on economics, and how the system deals with disturbances and infeasibilities. The ability of the multi-rate MHE to track the true states is also investigated.

Finally, the computational effort required to solve the MHE and shrinking horizon dynamic optimization problems is discussed.

7.1 Problem setup

The first-principles based EAF model described in Section 2 is used with the decision support system implementation as described in Section 6. The EAF operation considered here is a 60 minute batch process with 10 manipulated inputs. The process starts with an initial scrap charge, with a second scrap charge introduced at the 25th minute. Although the scrap charging times are considered to be fixed for the case studies, the charging times can also be optimized.

For the multi-tiered optimization algorithm, the max_{iter} parameter is set to 100. For Tier 2 of the optimization, the maximum allowed extension was chosen as 3. This is reasonable since it is not desirable to extend a EAF batch by a substantial amount in usual scenarios. The warm start procedure for the MHE and economic optimization problems is applied as described in Section 6.2.

Plant-model mismatch was incorporated by decreasing a power factor parameter k_p by 10% in the model employed by the multi-tiered optimization and MHE. This reduction in the k_p value leads to a significant reduction in the electrical energy transfered to the scrap Q_{arc} by the arc power P_r. The exact model equation affected by this mismatch is[7] $Q_{arc} = k_p P_r$. This mismatch was applied to all case studies. Another disturbance is introduced through a step increase of parameter k_{dm} in the plant during the batch operation. This change significantly reduces the melting rate of solid scrap, $\dot{M}_{melt}$, in the plant since the melt rate is modeled

using the following equation

$$\dot{M}_{melt} = \frac{Q_{ss}(T_{ss}/T_{melt})}{\left[\Delta H_{f,Fe} + \int_{T_{ss}}^{T_{melt}} C_{p,Fe} dt\right] k_{dm}}, \tag{24}$$

where T_{ss} and T_{melt} represent the solid-scrap temperature and melting point temperature of scrap. Q_{ss} denotes heat of fusion of Fe, and $C_{p,Fe}$ is the heat capacity of Fe. k_{dm} accounts for variation in the solid-scrap composition and its bulk density. To mitigate the effects of the mismatch and the disturbance, two disturbance states are introduced for the state variables representing the solid scrap mass and the moles of managenese in the *slag-metal* zone. This is implemented by using $\mathbf{B}_d$ in eq 23a as $[\mathbf{e}_i, \mathbf{e}_j]$, where i and j denote the state variable numbers for the solid scrap mass and the moles of managenese respectively in the state variable vector of size n, and $\mathbf{e}_i \in \mathbb{R}^n$ represents the ith unit vector. We chose $\mathbf{Q}_d$ (defined in Section 5.4) as 0.015 and 0.2 for disturbance states of the solid scrap mass and the moles of managenese respectively.

The measurements are available at different sampling rates as indicated in Table 1. There

Table 1: Multi-rate measurement structure for the EAF process.

Time (min)	0 . . . 42	43	44 . . . 46	47	48 . . . 60
Number of measured variables	6	**13**	6	**8**	6

are 6 fast measurements, comprising the off-gas compositions and the furnace roof and wall temperatures, that are available every 1 minute. Slag chemistry information in the form of 5 measurements is available at the 43rd minute of the batch duration. Molten metal carbon content and temperature are measured at 43rd and 47th minute. Since these slow measurements give valuable information about the process, a multi-rate MHE design is essential for optimal performance of the state estimator. The measurements with their corresponding sampling time points and variances are listed in Table 2. The MHE performance is demon-

strated when the initial conditions are unknown and there is plant-model mismatch as well as measurement noise.

Table 2: Measurement availability for the EAF process.

Measurement	Sampling Time	Variance
Off-gas compositions (CO, CO_2, O_2, H_2) (dimensionless)	Every 1 min	0.01
T_{roof}, T_{wall} (K)	Every 1 min	3
Slag compositions (FeO, Al_2O_3, SiO_2, MgO, CaO) (dimensionless)	43rd min	0.1
Molten-metal temperature (K)	43rd & 47th min	5
Molten-metal carbon content (dimensionless)	43rd & 47th min	0.01

Carrying out an observability analysis is important before proceeding with implementing MHE. This is particularly useful for systems where limited measurements are available. The DAE model was linearized at each sampling time and the observability metric value checked [45] at the corresponding time points. The metric value for a state variable represents the projection of its original value to the observable canonical space with the weight assigned being the associated singular value of the observability matrix. Ji and Rawlings [45] remark that very low values of the observability metric imply an unobservable or a weakly observable system. The lowest observability metric value identified for our system is 7.0×10^{-7}, from which we infer that the system is fully observable. It is to be noted that instead of carrying out the linearization procedure, output from the IPOPT solver can be examined directly to assess the system observability. [58]

It is possible that implementation of either of the three solutions obtained from the three tiers cannot actually ensure that the state variable value $\mathbf{x}_{plant}(t_{f_{tier}})$ at the final time is within specified bounds. This is due to disturbances effecting the plant and insufficient re-optimization triggers. Since there is typically an associated external cost when the state

variables do not meet the end-point constraints, it is important to monitor $\mathbf{x}_{est}$ and consider extending the batch when estimated states $\mathbf{x}_{est}(t_{f_{tier}})$ at $t_{f_{tier}}$ are not within the specified upper and lower bounds. We note here that such an estimate of the current states is useful to operators even without the optimization calculation being triggered.

7.2 Case Study descriptions and results

The next subsections describe three case studies conducted to evaluate the performance of the real-time advisory system. We chose the update times in the case studies to explore the impact of the frequency and timing of the optimization calls on the overall plant economics, and also to demonstrate the different tiers of the real-time algorithm. The first case study demonstrates the economic benefit of carrying out an increasing number of re-optimizations, and demonstrates effectiveness of the advisory system when the plant is affected by a major disturbance. The second case study demonstrates the effectiveness of Tier 2 in handling end-point infeasibility. The third case study explores the use of all the three optimization tiers.

7.2.1 Case Study 1: Economic benefit of real-time advisory system execution

In this case we compare four scenarios: when the optimization in the advisory system is activated once, twice, three and five times by the EAF operators. Since, there is flexibility in the advisory system for operators to execute it when decision support is required, this case illustrates the scenarios where the number of executions can differ. This case study emulates a situation where the plant is affected by major disturbances while the batch process is evolving in time. The parameter k_{dm}, described earlier, was increased in the plant by 5% at 28th minute. This step increase leads to a significant decrease in the rate of scrap melting from the 28th minute onwards during the batch. An immediate impact is that

more solid scrap will be left at 60th minute if limited reoptimizations are triggered, and a batch extension will be required. Besides the disturbance, we also considered plant-model mismatch, unknown initial states and measurement noise for all the case studies.

Scenario 1 involves a single optimization carried out at the start of the batch, with the calculated input sequence applied to the plant. In Scenario 2, a second reoptimization is carried out at the 30th minute, whereas for Scenario 3, two reoptimization calculations are performed at the 30th and 40th minute respectively. Scenario 4 considers 4 reoptimization calculations at the 30th, 40th, 50th and 58th minute respectively. The spread of reoptimization triggers throughout the batch duration is useful in picking up disturbances which in turn increases the opportunity to make an impact. If we reoptimize only at the start then it can be difficult to capture a disturbance which occurs later during the heat.

For all the four scenarios, MHE is executed every minute to provide the state estimates when needed. The end-point constraint is implemented for solid scrap mass (m_{ss}) at 60th minute, with the final scrap remaining specified to be less than or equal to 1 kg (out of 118.35 tonnes scrap charged). The input bounds for [F_{CaO}, $F_{C_{lance}}$, $F_{2ndCharge}$, $F_{Dolomite}$, $F_{C_{charge}}$, $F_{CH_4,brnr}$] are $\pm$20% of the nominal values. Lower bounds for P and [$F_{O_2,Jetbox1}, F_{O_2,Jetbox2}, F_{O_2,Jetbox2}$] are considered as 60% and 40% of the nominal values respectively. Upper bounds for P and [$F_{O_2,Jetbox1}, F_{O_2,Jetbox2}, F_{O_2,Jetbox2}$] are specified as 180% and 120% of the nominal values.

The four scenarios are compared in Table 3. For this case study, the multi-tiered optimization algorithm always terminates at Tier 1. We observe that for Scenario 2 a 2.85% increase in profit is achieved compared to Scenario 1. Also, Scenario 3 has a 1.32% higher economic objective value compared to Scenario 2, and Scenario 4 has 1.06% greater profit compared to Scenario 3. Considering that approximately 24 batches are run per day, these percentage increases translate to significant increases in annual steelmaking profit. MHE is able to detect the occurrence of a major disturbance, which impacts the reoptimizations. The case

Table 3: Results for Case Study 2.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4
Times at which advisory system was called (min)	0	$0, 30$	$0, 30, 40$	$0, 30, 40, 50, 58$
Number of re-optimizations	0	1	2	4
Economic objective function value ($)	9,100	9,360	9,484	9,585
Actual scrap left at 60th minute (kg)	2.8	2.5	2.3	1.0
Estimated scrap left at 60th minute (kg)	2.0	2.0	1.8	0.9

study demonstrates a substantial economic benefit of more frequent reoptimizations. In the limit of reoptimizations carried out at every 1 minute, the implementation naturally extends to NMPC. Figure 4 shows how the calculated inputs differ for the four scenarios. We observe that Scenario 2, 3 and 4 use considerably less oxygen and electric arc power post 30 minutes during the batch operation, leading to a substantial decrease in operating cost. Scenario 4 has lower oxygen usage but slightly more of electric arc power usage compared to Scenario 3.

From Table 3, we observe that scrap left at 60th minute is less when more reoptimizations are carried out. Since more reoptimizations are triggered for scenario 4, the scrap left at 60th minute is within the upper bound of 1 kg. However, the scrap left at 60th min was more than 1 kg for all the other three scenarios due to the sustained disturbance and no reoptimization late in the batch. This case demonstrates that reoptimizations are important both for increasing profit and satisfying end-point constraints. Scenarios 1 to 3 would require extension of the batch length to melt the remaining scrap, which would lead to a delayed start of subsequent batches and also delays in downstream processes. Although we have not taken these costs in account within this paper, such considerations can lead to further reduction of the final profit for Scenarios 1 to 3.

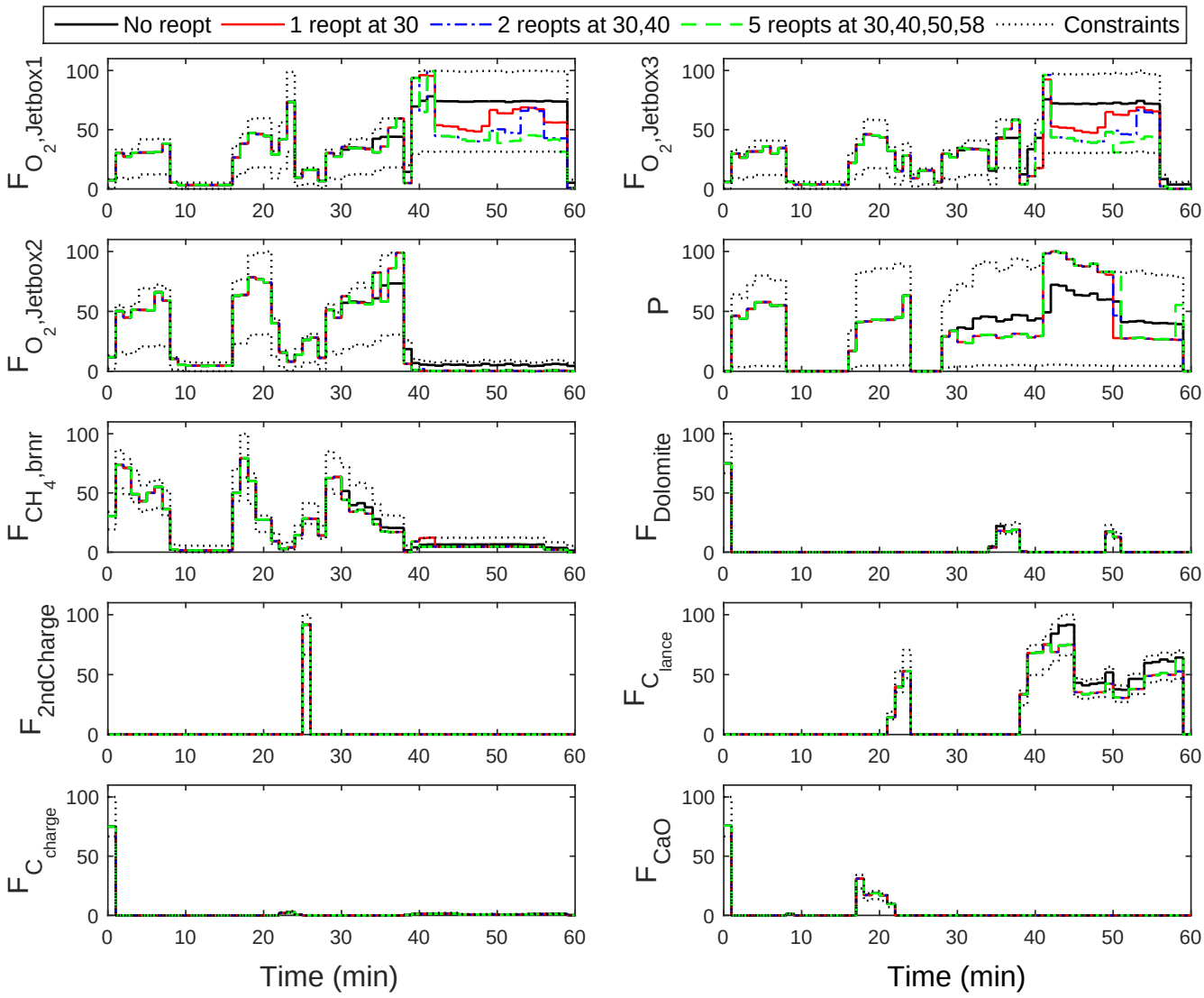

Figure 4: Input profiles for Case Study 1.

7.2.2 Case Study 2: Batch extension for achieving feasibility

In this case study, we explore the effectiveness of Tier 2 in the optimization sequence in achieving feasibility when the batch is extended by a sufficient time duration. The upper and lower bound for P are considered as 110% and 10% of the nominal values. Other input bounds are the same as for Case Study 1. Two reoptimizations are carried out at the 45th and 50th minute, respectively, of the batch process. The tight upper bound on P, plant-model mismatch and late reoptimizations led to Tier 1 solving up to maximum iteration limit of 100 without reaching an optimal solution at both the triggered time points. This led to Tier 2 being activated, where the maximum allowed extension is specified as 3 time steps. At the 45th and 50th minute, Tier 2 found that if the batch is extended by 3 and 2 time steps respectively then the scrap left will be within the upper bound of 1 kg. The minimum integral extension is found using the serial approach. The profit obtained is \$9,476 and the final scrap left at 62nd minute is 0.84 kg. This case study shows that the tiered approach is useful in extending the batch optimally for reducing the final scrap amount to within the specified limit.

7.2.3 Case Study 3: Feasibility through multi-tiered optimization strategy

In this case study, a single reoptimization is carried out at the 45th minute of the batch process. A disturbance is introduced through a step increase of k_{dm} by 12.3% at 38th minute of the batch operation. The lower and upper bounds of the electric power input were also reduced to 10% and 90% of the nominal values respectively. This combined change led to Tier 1 solving up to the maximum iteration limit of 100 without finding an optimal solution. This activated Tier 2, and three extended dynamic optimization problems were solved serially for each of the integral extensions up to 3 time steps (corresponding to the specified extension limit). Since all the three problems reached the maximum iteration limit,

the advisory system moved to Tier 3 to solve a relaxed problem by softening the end point constraint on final solid scrap mass. In this case study, we set the tuning parameter D as 0.01. With the relaxed extended optimization problem solved at the 45th minute, 4.5 kg of solid scrap was left at 63rd minute, yielding a profit of \$9,304. This case demonstrates that the tiered algorithm is able to compute a reasonable solution even when the original optimization problem is infeasible.

7.3 Moving Horizon Estimation results

In this section we discuss the performance of the multi-rate MHE scheme. For all the three case studies, Gaussian noise of 1% relative variance was added to the true initial conditions of the state variables. Measurements were perturbed by adding Gaussian noise of relative variances according to the values given in Table 2. The covariance matrix Q for the model uncertainty was chosen based on multiple trial simulations. We also implemented upper and lower bounds for the state, the algebraic and the model noise variables. A moving horizon length of 6 was selected for all the case studies. Figure 5 shows a subset of the true states and the estimated states for Case Study 3, as well as how the absolute value of estimation error, given by $|x_{true} - x_{estimated}|$, varies with respect to time. We observe that MHE is able to track the true states despite of the plant-model mismatch, initial state discrepancy, and measurement noise. The estimation error can be seen to increase around the 40-50th minute due to the disturbance being introduced in the plant. However, MHE handles this and decreases the estimation error post 45th min.

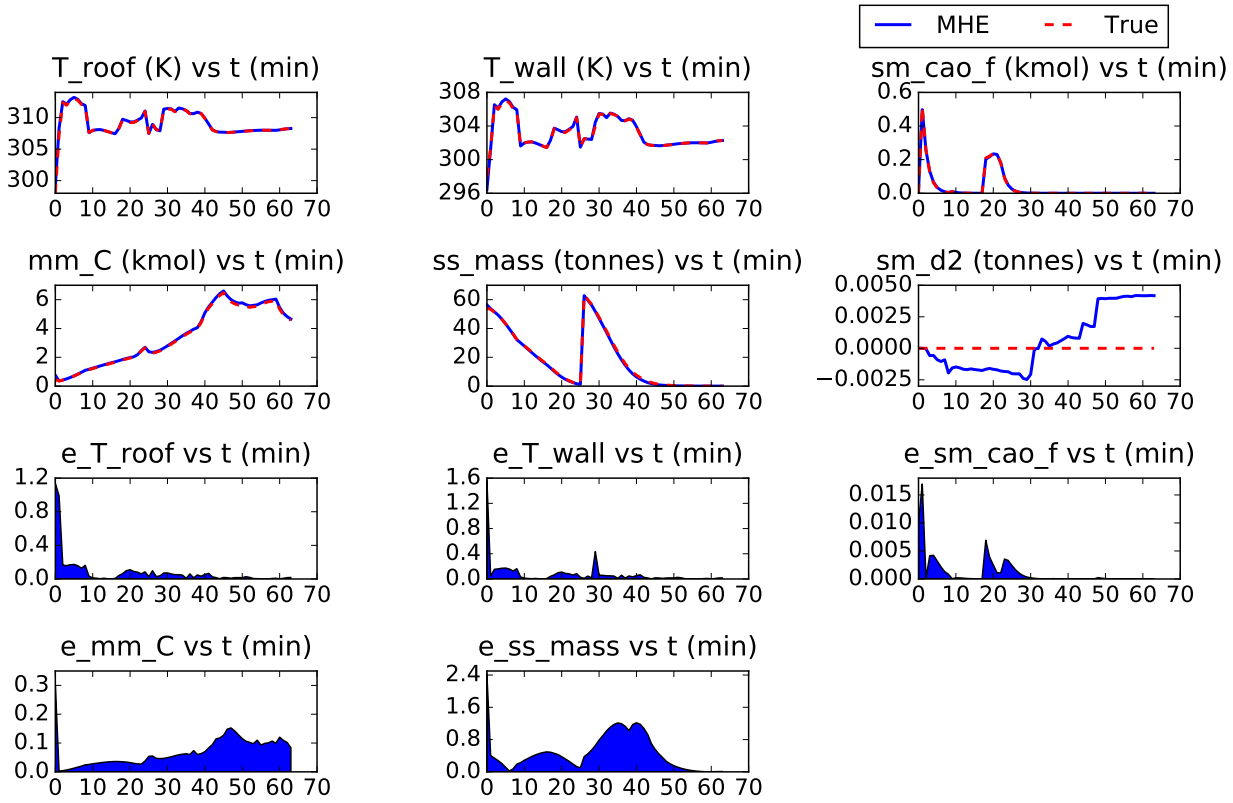

Figure 5: State estimate profiles and corresponding estimation error (variable name appended with 'e_') varying with respect to time (in minutes) for Case Study 3. T_roof (temperature of furnace roof), T_wall (temperature of furnace wall), sm_cao_f (floating carbon content in slag-metal zone), mm_C (moles of carbon in molten-metal zone), ss_mass (mass of solid scrap) and sm_d2 (disturbance state for ss_mass).

7.4 Computational results

An Intel Core i7-3770 processor (with 4 CPU cores) running Windows 7 at 3.4 GHz was employed for all numerical computations. The CPU times required for solving the multi-rate MHE problems, with (fMHE) and without (nMHE) the use of the proposed multi-tiered initialization scheme, for Case Study 3 are shown in Figure 6. The average solution times for fMHE and nMHE are 0.49 and 0.71 seconds respectively. Thus, there is 31% decrease in MHE solve time due to the use of the multi-tiered warm-start strategy. The solution times for the multi-tiered dynamic optimization problems of Case Study 3 are given in Table 4, where the advantage of using the multi-tiered initialization scheme is apparent. The solution times for the advisory system used in Case Study 1 and 2 were on average below 5 seconds. Furthermore, the solution time for the predicted problems solved in the multi-tiered initialization strategy for all the three case studies was well below 1 minute.

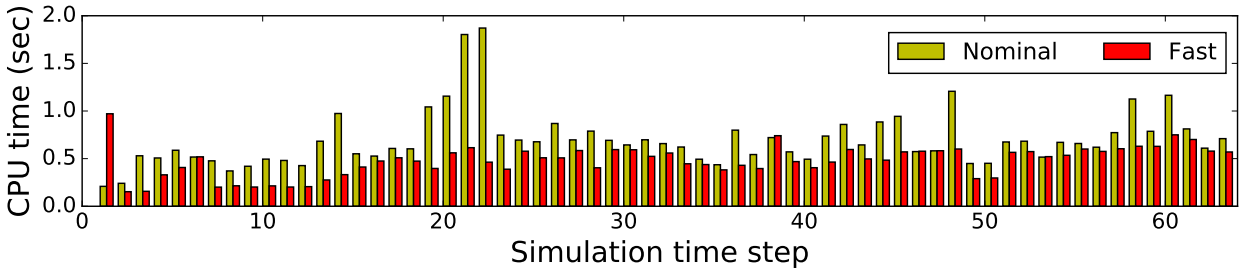

Figure 6: Solution times for multi-rate MHE problems (horizon length of 6 minutes) for Case Study 3. 'Fast' represents solution times when the multi-tiered initialization scheme is used. 'Nominal' denotes solutions times when the proposed initialization scheme is not utilized.

8 Conclusion

In this work, a real-time dynamic optimization-based advisory system for EAF batch operation was introduced. The advisory system combines multi-tiered dynamic optimization with MHE to provide online decision support to EAF plant operators. The three tiers of

Table 4: Solution times for the multi-tiered dynamic optimization problems of Case Study 3.

Time points where multi-tiered optimizer is called	0th minute	45th minute				
Activated tier number	1	1	2			3
Solution time (CPU seconds) without the use of multi-tiered initialization	25.7	6.8	6.6	7.3	7.1	2.1
Solution time (CPU seconds) with the use of multi-tiered initialization	2.1	1.1	0.9	1.2	1.1	0.6

the optimizer effectively handle the endpoint constraints. At any point during the batch operation, the optimizer can be triggered by the operators. The initial conditions for the optimization problem are obtained from multi-rate MHE which runs at every sampling instance. A multi-tiered initialization scheme was also proposed for obtaining fast solutions from the advisory system.

Case studies for the electric arc steelmaking process demonstrated the benefit of using the real-time advisory system. The usefulness of reoptimization in improving the economic performance and mitigating end-point constraint violations were shown. The strong convergence of MHE directly influenced the optimizer results and provided a firm basis for activating the tiered optimization algorithm. Computation times demonstrated the viability of the advisory system in providing decision support to operators in real-time.

A useful avenue for future work is exploration of a process monitoring layer that could automate the triggering process, or that could advise the operator when to trigger the optimization.

Acknowledgement

Support from the McMaster Steel Research Centre (SRC) and the McMaster Advanced

Control Consortium (MACC) for this work is gratefully acknowledged.

References

1. Steel Statistical Yearbook of World Steel Association, http://www.worldsteel.org/steel-by-topic/statistics/steel-statistical-yearbook-.html. 2016.

2. Fruehan, R. J. *The Making, Shaping, and Treating of Steel*, 11th ed.; AISE Steel Foundation, 1998.

3. Woodside, C.; Pagurek, B.; Pauksens, J.; Ogale, A. Singular arcs occurring in optimal electric steel refining. *IEEE Transactions on Automatic Control* **1970**, *15*, 549–556.

4. Oosthuizen, D.; Craig, I.; Pistorius, P. Model predictive control of an electric arc furnace off-gas procedure combined with temperature control. Africon, 1999 IEEE. 1999; pp 415–420.

5. Bekker, J. G.; Craig, I. K.; Pistorius, P. C. Modeling and simulation of an electric arc furnace process. *ISIJ International* **1999**, *39*, 23–32.

6. Matson, S.; Ramirez, W. F. Optimal operation of an electric arc furnace. 57 th Electric Furnace Conference. 1999; pp 719–730.

7. MacRosty, R. D. M.; Swartz, C. L. E. Dynamic modeling of an industrial electric arc furnace. *Industrial & Engineering Chemistry Research* **2005**, *44*, 8067–8083.

8. MacRosty, R. D. M.; Swartz, C. L. E. Dynamic optimization of electric arc furnace operation. *AIChE Journal* **2007**, *53*, 640–653.

9. MacRosty, R. D. M.; Swartz, C. L. E. Nonlinear predictive control of an electric arc furnace. *IFAC Proceedings Volumes* **2007**, *40*, 285–290.

10. Rashid, M. M.; Mhaskar, P.; Swartz, C. L. E. Multi-rate modeling and economic model predictive control of the electric arc furnace. *Journal of Process Control* **2016**, *40*, 50–61.

11. Rashid, M. M.; Mhaskar, P.; Swartz, C. L. E. Handling multi-rate and missing data in variable duration economic model predictive control of batch processes. *AIChE Journal* **2017**, *63*, 2705–2718.

12. Shyamal, S.; Swartz, C. L. E. Real-time energy management for electric arc furnace operation. *Journal of Process Control* **2018**, *https://doi.org/10.1016/j.jprocont.2018.03.002*.

13. Prasad, V.; Schley, M.; Russo, L. P.; Bequette, B. W. Product property and production rate control of styrene polymerization. *Journal of Process Control* **2002**, *12*, 353–372.

14. Qu, C. C.; Hahn, J. Process monitoring and parameter estimation via unscented Kalman filtering. *Journal of Loss Prevention in the Process Industries* **2009**, *22*, 703–709.

15. Robertson, D. G.; Lee, J. H. On the use of constraints in least squares estimation and control. *Automatica* **2002**, *38*, 1113–1123.

16. Rao, C. V.; Rawlings, J. B.; Mayne, D. Q. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control* **2003**, *48*, 246–258.

17. Liu, J. Moving horizon state estimation for nonlinear systems with bounded uncertainties. *Chemical Engineering Science* **2013**, *93*, 376–386.

18. Allgöwer, F.; Badgwell, T. A.; Qin, J. S.; Rawlings, J. B.; Wright, S. J. Nonlinear predictive control and moving horizon estimation-an introductory overview. In *Advances in Control*; Springer, 1999; pp 391–449.

19. Billings, S.; Boland, F.; Nicholson, H. Electric arc furnace modelling and control. *Automatica* **1979**, *15*, 137–148.

20. Ghobara, Y. *Modeling, Optimization and Estimation in Electric Arc Furnace (EAF) Operation*; Masters thesis, McMaster University, 2013.

21. Shyamal, S.; Swartz, C. L. E. A Multi-rate moving horizon estimation framework for electric arc furnace operation. *IFAC-PapersOnLine* **2016**, *49*, 1175–1180.

22. Flores-Cerrillo, J.; MacGregor, J. F. Control of particle size distributions in emulsion semibatch polymerization using mid-course correction policies. *Industrial & Engineering Chemistry Research* **2002**, *41*, 1805–1814.

23. Biegler, L. T. *Nonlinear Pogramming: Concepts, Algorithms, and Applications to Chemical Processes*; SIAM, 2010.

24. Li, Y.; Fruehan, R. J. Computational fluid-dynamics simulation of postcombustion in the electric-arc furnace. *Metallurgical and Materials Transactions B* **2003**, *34*, 333–343.

25. Tang, X.; Kirschen, M.; Abel, M.; Pfeifer, H. Modelling of EAF off-gas post combustion in dedusting systems using CFD methods. *Steel Research International* **2003**, *74*, 201–210.

26. Irons, G. A. Developments in electric arc furnace steelmaking. Proceedings, AISTech. 2005; pp 3–21.

27. Cameron, A.; Saxena, N.; Broome, K. Optimizing EAF operations by dynamic process simulation. Proceedig of the 56th Electric Furnace Conference. 1998; pp 689–696.

28. Matson, S.; Ramirez, W. F. Optimal operation of an electric arc furnace. 57th Electric Furnace Conference. 1999; pp 719–730.

29. Modigell, M.; Trabert, A.; Monheim, P. Modelling technique for metallurgical processes and its applications. *AISE Steel Technol.* **2001**, *28*, 45–47.

30. Process Systems Enterprise Ltd., gPROMS, www.psenterprise.com/gproms. 2015.

31. Andersson, J. A General-Purpose Software Framework for Dynamic Optimization. PhD thesis, Arenberg Doctoral School, KU Leuven, Department of Electrical Engineering (ESAT/SCD) and Optimization in Engineering Center, Kasteelpark Arenberg 10, 3001-Heverlee, Belgium, 2013.

32. Cao, Y.; Swartz, C. L.; Flores-Cerrillo, J.; Ma, J. Dynamic modeling and collocation-based model reduction of cryogenic air separation units. *AIChE Journal* **2016**, *62*, 1602–1615.

33. Hahn, J.; Edgar, T. F. Balancing approach to minimal realization and model reduction of stable nonlinear systems. *Industrial & Engineering Chemistry Research* **2002**, *41*, 2204–2212.

34. Magnusson, F. Numerical and symbolic methods for dynamic optimization. PhD thesis, Lund University, Department of Automatic Control, Lund Institute of Technology, Lund University, 2016.

35. Åkesson, J.; Årzén, K.-E.; Gäfvert, M.; Bergdahl, T.; Tummescheit, H. Modeling and optimization with Optimica and JModelica.org-Languages and tools for solving large-scale dynamic optimization problems. *Computers & Chemical Engineering* **2010**, *34*, 1737–1749.

36. Chachuat, B.; Singer, A. B.; Barton, P. I. Global mixed-integer dynamic optimization. *AIChE Journal* **2005**, *51*, 2235–2253.

37. Bansal, V.; Sakizlis, V.; Ross, R.; Perkins, J. D.; Pistikopoulos, E. N. New algorithms for mixed-integer dynamic optimization. *Computers & Chemical Engineering* **2003**, *27*, 647–668.

38. Maciejowski, J. M. *Predictive Control with Constraints*; Pearson Education, 2002.

39. Swartz, C. L. E. An algorithm for hierarchical supervisory control. *Computers & Chemical Engineering* **1995**, *19*, 1173–1180.

40. Chong, Z.; Swartz, C. L. E. Optimal operation of process plants under partial shutdown conditions. *AIChE Journal* **2013**, *59*, 4151–4168.

41. Chong, Z.; Swartz, C. L. E. Optimal response under partial plant shutdown with discontinuous dynamic models. *Computers & Chemical Engineering* **2016**, *86*, 120–135.

42. Jang, H.; Lee, J. H.; Biegler, L. T. A robust NMPC scheme for semi-batch polymerization reactors. *IFAC-PapersOnLine* **2016**, *49*, 37–42.

43. Krämer, S.; Gesthuisen, R.; Engell, S. Fixed structure multirate state estimation. American Control Conference (ACC). 2005; pp 4613–4618.

44. López-Negrete, R.; Biegler, L. T. A moving horizon estimator for processes with multirate measurements: A nonlinear programming sensitivity approach. *Journal of Process Control* **2012**, *22*, 677–688.

45. Ji, L.; Rawlings, J. B. Application of MHE to large-scale nonlinear processes with delayed lab measurements. *Computers & Chemical Engineering* **2015**, *80*, 63–72.

46. Rao, C. V.; Rawlings, J. B. Constrained process monitoring: Moving-horizon approach. *AIChE Journal* **2002**, *48*, 97–109.

47. Alessandri, A.; Baglietto, M.; Battistelli, G. Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes. *Automatica* **2008**, *44*, 1753–1765.

48. Zavala, V. M. Stability analysis of an approximate scheme for moving horizon estimation. *Computers & Chemical Engineering* **2010**, *34*, 1662–1670.

49. Lopez-Negrete, R.; Patwardhan, S. C.; Biegler, L. T. Constrained particle filter approach to approximate the arrival cost in moving horizon estimation. *Journal of Process Control* **2011**, *21*, 909–919.

50. Qu, C. C.; Hahn, J. Computation of arrival cost for moving horizon estimation via unscented Kalman filtering. *Journal of Process Control* **2009**, *19*, 358–363.

51. Diehl, M.; Ferreau, H. J.; Haverbeke, N. Efficient numerical methods for nonlinear MPC and moving horizon estimation. In *Nonlinear Model Predictive Control*; Springer, 2009; pp 391–417.

52. Biegler, L. T.; Yang, X.; Fischer, G. A. G. Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization. *Journal of Process Control* **2015**, *30*, 104–116.

53. Zavala, V. M.; Laird, C. D.; Biegler, L. T. A fast moving horizon estimation algorithm based on nonlinear programming sensitivity. *Journal of Process Control* **2008**, *18*, 876–884.

54. Diehl, M.; Findeisen, R.; Allgöwer, F.; Bock, H. G.; Schlöder, J. P. Nominal stability of real-time iteration scheme for nonlinear model predictive control. *IEE Proceedings-Control Theory and Applications* **2005**, *152*, 296–308.

55. DeHaan, D.; Guay, M. A new real-time approach for nonlinear model predictive control. *IFAC Proceedings Volumes* **2005**, *38*, 1007–1012.

56. Wächter, A.; Biegler, L. T. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* **2006**, *106*, 25–57.

57. Hindmarsh, A. C.; Brown, P. N.; Grant, K. E.; Lee, S. L.; Serban, R.; Shumaker, D. E.;

Woodward, C. S. SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Transactions on Mathematical Software (TOMS)* **2005**, *31*, 363–396.

58. Zavala, V. M.; Biegler, L. T. Optimization-based strategies for the operation of low-density polyethylene tubular reactors: Moving horizon estimation. *Computers & Chemical Engineering* **2009**, *33*, 379–390.

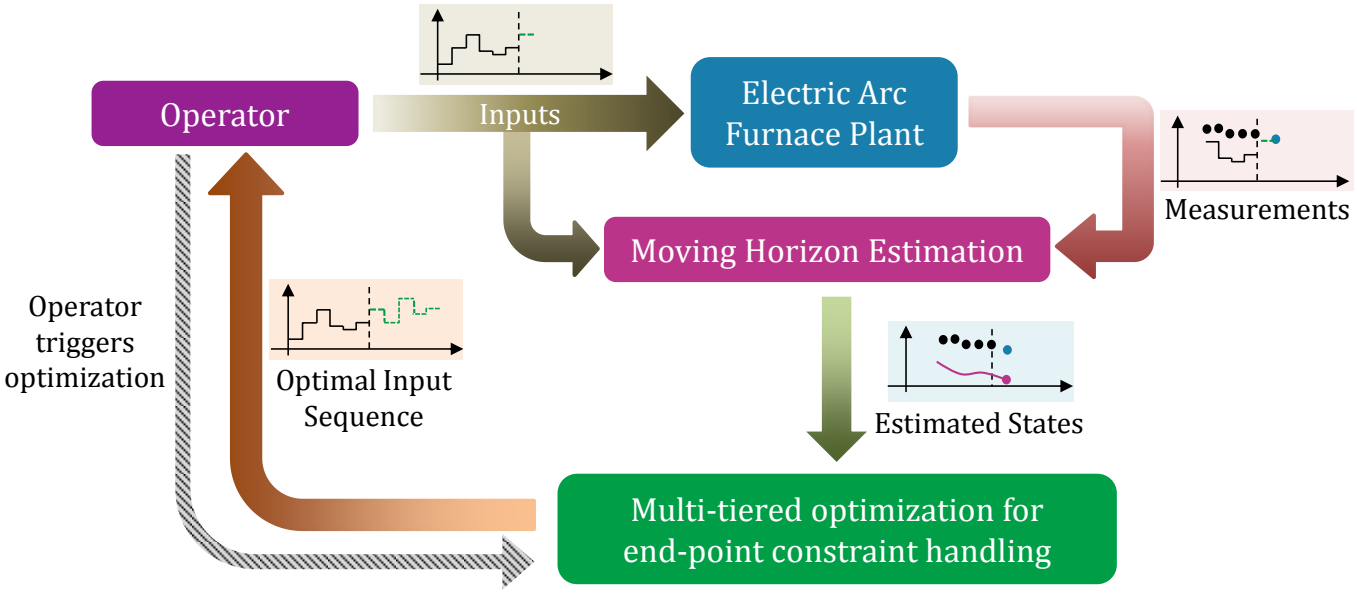

Figure 7: For Table of Contents Only.

List of Figures