An Augmented Lagrangian Filter Method for Real-Time Embedded Optimization

Nai-Yuan Chiang, Rui Huang, and Victor M. Zavala, *Member, IEEE*

***Abstract*—We present a filter line-search algorithm for nonconvex continuous optimization that combines an augmented Lagrangian function and a constraint violation metric to accept and reject steps. The approach is motivated by real-time optimization applications that need to be executed on embedded computing platforms with limited memory and processor speeds. The proposed method enables primal–dual regularization of the linear algebra system that in turn permits the use of solution strategies with lower computing overheads. We prove that the proposed algorithm is globally convergent and we demonstrate the developments using a nonconvex real-time optimization application for a building heating, ventilation, and air conditioning system. Our numerical tests are performed on a standard processor and on an embedded platform. We demonstrate that the approach reduces solution times by a factor of over 1000.**

***Index Terms*—Control, embedded, nonconvex, optimization, real time.**

I. Motivation

REAL-TIME optimization applications such as model predictive control (MPC), state estimation, signal processing, and set-point optimization rely on fast and repetitive solutions of optimization problems [1]. Solution time needs to be reduced in order to capture fast dynamic behavior and high-frequency disturbances (in some applications it is necessary to solve problems in microseconds). In some applications (e.g., smart thermostats, unmanned vehicles) it is also necessary to solve optimization problems on a low-cost embedded computing platform operating with limited processing speed, memory, and power. Limitations imposed by embedded platforms limit the use of off-the-self optimization algorithms and software implementations.

A notable application of real-time optimization on embedded platforms is building energy management. Over 50% of the energy consumption of a building is used by heating, ventilation, and air conditioning (HVAC) systems [2]. An HVAC system is composed of different subsystems such as air handling units (AHU), fans, ducts, and variable air volume (VAV) boxes that condition and deliver air to multiple building zones. A real-time HVAC set-point optimization application identifies feasible and cost-efficient resource allocations that satisfy subsystem constraints as disturbances such as weather and internal loads vary in time [2]. Many of these optimization applications require fast solutions of nonlinear optimization problems to capture, for instance, changes in occupancy or faults [3].

Interior-point algorithms provide a flexible framework to solve optimization problems on a variety of computing platforms as they enable modularity of linear algebra implementations [4]. Among different algorithmic variants, filter line-search strategies have proven to be particularly flexible and robust for handling complex applications. Software implementations for general nonconvex problems include `IPOPT` [5], `IPFILTER` [5], and `PIPS-NLP` [6]. As argued in [6], a filter line-search setting enables the implementation of a wider range of linear algebra strategies (to solve the primal–dual linear system) compared to trust-region counterparts. Such flexibility enables implementations on diverse computer architectures that range from large-scale computing clusters to embedded computing platforms [7]–[9]. In some MPC applications, for instance, it is possible to exploit the linear algebra structure of the optimization model to accelerate solutions or to tailor the linear algebra strategy to a specific platform [8], [10], [11]. In nonconvex settings, a filter line-search algorithm monitors for the presence of negative curvature by using inertia information from the linear algebra system [5]. If negative curvature is detected, the Hessian matrix is regularized (called primal regularization) to ensure that the Newton step delivers descent directions. The algorithm might also need to perform dual regularization whenever the Jacobian matrix is rank deficient. Dual regularization has proven to be an effective strategy in actual implementations [12], [13] but no formal theoretical treatment has been provided. In particular, global convergence of existing filter line-search methods cannot be guaranteed in the presence of dual regularization.

Another important class of methods that are attractive for real-time optimization and control are augmented Lagrangian (AL) methods [14]–[17]. These methods are attractive because they operate only on the primal space (thus reducing computational complexity) and directly minimize a merit function. Moreover, using AL schemes within an interior-point framework enable modularity in linear algebra (because the linear algebra kernel

Manuscript received January 17, 2017; revised January 18, 2017 and March 31, 2017; accepted April 11, 2017. Date of publication April 17, 2017; date of current version December 1, 2017. The work of V. M. Zavala was supported by the U.S. Department of Energy under Grant DE-SC0014114. Recommended by Associate Editor W. Michiels. *(Corresponding author: Victor M. Zavala.)*

N.-Y. Chiang and R. Huang are with United Technologies Research Center, East Hartford, CT 06118, USA (e-mail: chiangn@utrc.utc.com; huangr76@gmail.com).

V. M. Zavala is with the Department of Chemical and Biological Engineering, University of Wisconsin-Madison, Madison, WI 53706, USA (e-mail: victor.zavala@wisc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2017.2694806

does not change and can be solved by using conjugate gradient (CG) schemes and Cholesky factorization approaches with static structures). AL methods can also be implemented within an active-set framework by using projection methods combined with CG and Cholesky but such approaches limit modularity in linear algebra because the structure of the linear algebra kernel is not static [18]–[20]. Consequently, from a modularity standpoint, interior-point frameworks are preferred. A drawback of using AL within an interior-point framework, however, is that AL algorithms require the solution of nested subproblems (AL subproblem within the barrier problem). In summary, a filter line-search approach provides more flexibility to accept/reject steps and accelerate convergence but steps are more expensive to compute.

In this work, we present a filter line-search strategy for nonconvex optimization that uses a constraint violation metric and an AL function to reject and accept steps (as opposed to a constraint violation metric and the objective function used in a standard setting). We refer to this strategy as the AL filter line-search algorithm. We show that, by designing the filter in this form, it is possible to guide primal–dual regularization of the linear system. In particular, we prove that the proposed algorithm is globally convergent even when the linear system is persistently regularized in the primal–dual space. We thus provide a solid theoretical support to address curvature and rank deficiency issues. More importantly, by using dual regularization, we can inherit computational features of AL approaches that are attractive in embedded implementations. In particular, we can use a fast Schur–Cholesky decomposition scheme that operates on the primal space. With this, we avoid the need to use symmetric indefinite factorizations operating on the primal–dual space (which involve higher overheads and require more sophisticated computing processors). We implement our strategy on the interior-point solver `PIPS-NLP` and perform numerical tests by solving HVAC optimization problems and on an embedded computing platform. We demonstrate that the proposed approach reduces solution times by a factor of over 1000.

The paper is structured as follows. In Section II, we present the AL filter line-search algorithm. The convergence analysis of the scheme is presented in Section III. In Section IV, we present numerical results for the HVAC application. We close the paper with a short conclusion and directions for future work.

II. Algorithmic Development

A. Basic Notation and Preliminary Results

We are interested in the solution of problems of the form:

$$\min_{x} \; f(x) \quad \text{s.t.} \quad c(x) = 0 \tag{1}$$

where $x \in \mathbb{R}^n$ are the primal variables and we make the blanket assumption that the mappings $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m$ are twice differentiable on an open set $\mathcal{C} \subseteq \mathbb{R}^n$. In addition, we assume that their values and derivatives are bounded in $\mathcal{C}$. To handle NLPs with inequality constraints, we use a logarithmic barrier framework, which transforms the NLP into a sequence of equality constrained NLPs of the form (1). The Lagrange function of (1) is given by

$$L(x, y) := f(x) + y^T c(x) \tag{2}$$

where $y \in \mathbb{R}^m$ denote the Lagrange multipliers (dual variables) of the equality constraints. We define the constraint violation measure:

$$h(x) := \frac{1}{2}\|c(x)\|^2 = \frac{1}{2}c(x)^T c(x) \tag{3}$$

where $\|\cdot\|$ is the Euclidean norm. The AL function is given by

$$AL(x, y, \delta^c) := f(x) + y^T c(x) + \delta^c \cdot h(x) \tag{4}$$

where $\delta^c \geq 0$ is a *fixed* penalty parameter and we note that $L(x, y) = AL(x, y, 0)$. Since δ^c is fixed, we drop the dependence of the AL function on this parameter from the notation. We say that an iterate (x_k, y_k) is feasible if $c(x_k) = 0$ and stationary if $\nabla_x L(x_k, y_k) = 0$. We also recall that

$$\begin{aligned}\nabla_x AL(x, y) &= \nabla_x L(x, y) + \delta^c \cdot \nabla_x c(x) c(x) \\ &= \nabla_x f(x) + \nabla_x c(x)(y + \delta^c \cdot c(x)).\end{aligned} \tag{5}$$

Consequently, if (x_k, y_k) is both feasible and stationary, we have that $\nabla_x AL(x_k, y_k) = 0$.

We develop a *regularized* Newton method that computes a step (d_k, p_k) at iteration k for primal and dual variables. This is done by solving the following linear system (we refer to this as the KKT or augmented system):

$$\begin{bmatrix} W_k + \delta_k^w I & A_k^T \\ A_k & -\frac{1}{\delta_k^c} I \end{bmatrix} \begin{pmatrix} d_k \\ p_k \end{pmatrix} = -\begin{pmatrix} \nabla L_k \\ c_k \end{pmatrix}. \tag{6}$$

For clarity in the presentation, we write the second row as

$$A_k d_k + c_k = \frac{1}{\delta_k^c} p_k. \tag{7}$$

We refer to the matrix on the left-hand side as the KKT matrix. Here, $W_k \in \mathbb{R}^{n \times n}$ is a symmetric matrix that approximates the Hessian of the Lagrangian $\nabla_{xx} L(x_k, y_k)$ and we assume W_k to be bounded on the open set $\mathcal{C}$. The Jacobian of the constraints is denoted as $A_k := \nabla_x c(x_k)^T \in \mathbb{R}^{m \times n}$ and we define the quantities $\nabla L_k := \nabla_x L(x_k, y_k) = g_k + A_k^T y_k$, $g_k := \nabla_x f(x_k)$, $c_k := c(x_k)$, $\nabla AL_k := \nabla_x AL(x_k, y_k, \delta^c) = g_k + A_k^T(y_k + \delta^c c_k)$, and $\nabla h_k := \nabla_x h(x_k) = A_k^T c_k$.

Parameters $\delta_k^w, \delta_k^c \geq 0$ are *primal and dual regularization parameters*, respectively, that are adjusted at each iteration k. We highlight that δ_k^c and δ^c are parameters that are related but they are not necessarily the same (the first one might or might not be updated at each iteration k to regularize the linear system while the second one remains fixed). We use the same notation, however, to highlight their relationship.

We pick δ_k^w, δ_k^c, such that the *KKT matrix is nonsingular at each iteration*. We refer to this test on the KKT matrix as the *nonsingularity test* (NST). From (7), we can see that the regularization of the 2×2 block of the KKT matrix introduces a perturbation to the linearized constraints. To control the size of such perturbation, we define the *dual perturbation test* (DPT)

$$\frac{1}{\delta_k^c}\|p_k\| \leq \beta \, \|c_k\| \tag{8}$$

where $\beta \in [0, 1)$. We see that adding dual regularization to the linear system has the effect of pushing the search step away from satisfying the linearized constraint violation. The perturbation vanishes for the unregularized system with $1/\delta_k^c = 0$. We also note that, if $h_k = 0$ (i.e., the iterate x_k is feasible), then DPT can only be guaranteed to hold if $1/\delta_k^c = 0$.

We also impose the following *curvature test* (CT) for step d_k:

$$d_k^T(W_k + \delta_k^w I)d_k \geq \kappa d_k^T d_k \tag{9}$$

for $\kappa > 0$. This test is necessary to ensure that the Newton step is a suitable descent direction for the AL [13].

To see that there exists a regularization pair (δ_k^w, δ_k^c) such that NST together with DPT and CT can hold (at all iterates with $h_k > 0$) we define the Schur system of (6):

$$S_k d_k = -\nabla L_k - \delta_k^c A_k^T c_k \tag{10}$$

where S_k is the *Schur complement matrix* and has the form:

$$S_k := W_k + \delta_k^w I + \delta_k^c A_k^T A_k. \tag{11}$$

The Schur system representation exists if $\frac{1}{\delta_k^c} > 0$ and is obtained by eliminating the dual step p_k. Moreover, if $\frac{1}{\delta_k^c} > 0$ holds, the Schur complement theorem guarantees that the KKT matrix is nonsingular if and only if the Schur matrix S_k is nonsingular. In particular, if $\frac{1}{\delta_k^c} > 0$, then S_k exists and, if A_k has full row rank, then $\delta_k^c A_k^T A_k$ is positive definite for any δ_c^k. Consequently, in such a case, we can choose any δ_k^c such that the DPT (8) holds and we only need to choose δ_k^w large enough such that $\delta_k^w + \lambda_{\min}(W_k) > 0$ to guarantee nonsingularity of the KKT matrix. If $\frac{1}{\delta_k^c} = 0$ and A_k has full row rank, then we pick δ_k^w such that the reduced Hessian $Z_k^T(W_k + \delta_k^w I)Z_k$ (with Z_k spanning the null-space of A_k) is nonsingular and therefore the KKT matrix is nonsingular (see [13, Lemma 3.1]) and such that CT holds. If A_k does not have full row rank, then we can make δ_k^c large enough such that DPT holds and, for this choice of δ_k^c, we pick sufficiently large δ_w^k such that $\delta_k^w + \lambda_{\min}(W_k + \delta_k^c A_k^T A_k) > 0$ holds to ensure that the KKT matrix is nonsingular (NST holds) and that CT holds. We thus conclude that we can always find a regularization pair (δ_k^w, δ_k^c), such that NST, CT, and DPT hold.

We highlight that, if the Jacobian A_k has full row rank, we can ensure that NST and CT hold by picking δ_k^w large enough, such that the reduced Hessian is positive definite. As shown in [13], however, this approach is restrictive in the sense that it requires the Jacobian to be full rank and it requires inertia information of the KKT matrix to infer if the reduced Hessian is positive definite. By ensuring that NST and CT hold directly, we bypass these limitations.

The Schur complement representation also reveals that one can compute the step d_k by solving system (10) and then recover the multiplier step p_k from (7). This is important because the Schur complement has the dimension of the primal space (as opposed to the KKT matrix which has the dimension of the primal-dual space). It is also well known that the Schur matrix is the Hessian of the AL and can be made positive definite by properly selecting (δ_k^w, δ_k^c). This provides flexibility to use different linear algebra strategies such as Cholesky factorization or a preconditioned conjugate gradient scheme, as opposed to symmetric indefinite factorizations that operate on the full primal–dual KKT system (6). This flexibility is desirable for implementations on embedded platforms that have limited computing resources.

We emphasize that, at any feasible iterate (with $h_k = 0$), we must ensure that the Jacobian has full row rank (so that NST and DPT hold.) This implies that the proposed algorithm will need to revert to the full augmented system representation with $1/\delta_k^c = 0$ whenever we encounter a feasible iterate. We elaborate on this issue later on.

We define the following *criticality measure*:

$$\chi_k := \|d_k\|. \tag{12}$$

We will show that, under the globalization framework considered, this can be used as a measure of stationarity. We also define the following linear models of the difference between the constraint violation and of the AL from point (x_k, y_k) to point $(x_k + \alpha d_k, y_k + \alpha p_k)$ (where $\alpha \geq 0$ is the step size):

$$m_k^h(\alpha) := \alpha c_k^T A_k d_k \tag{13a}$$

$$\begin{aligned} m_k^{AL}(\alpha) := {} & \alpha g_k^T d_k + \alpha y_k^T A_k d_k \\ & + \alpha p_k^T(c_k + \alpha A_k d_k) + \alpha\delta^c c_k^T A_k d_k. \end{aligned} \tag{13b}$$

The first relationship is easy to establish, while the second relationship follows from

$$\begin{aligned} & AL(x_k + \alpha d_k, y_k + \alpha p_k) - AL(x_k, y_k) \\ & = f(x_k + \alpha d_k) - f(x_k) \\ & \quad + (y_k + \alpha p_k)^T c(x_k + \alpha d_k) - y_k^T c(x_k) \\ & \quad + \delta^c h(x_k + \alpha d_k) - \delta^c h(x_k) \\ & = \alpha g_k^T d_k + O(\alpha^2 ||d_k||^2) \\ & \quad + (y_k + \alpha p_k)^T(c_k + \alpha A_k d_k + O(\alpha^2 ||d_k||^2)) \\ & \quad - y_k^T c_k + \alpha\delta^c c_k^T A_k d_k + O(\alpha^2 ||d_k||^2) \\ & = m_k^{AL}(\alpha) + O(\alpha^2 ||d_k||^2). \end{aligned} \tag{14}$$

We say that the primal step d_k is a *descent direction* for the constraint violation if $m_k^h(\alpha) < 0$ and that (d_k, p_k) is a descent direction for the AL if $m_k^{AL}(\alpha) < 0$.

From (13) and (14), it is clear that the following bounds hold:

$$|h(x_k + \alpha d_k) - h_k - m_k^h(\alpha)| \leq M_h \alpha^2 \|d_k\|^2 \tag{15a}$$

$$\begin{aligned} & |AL(x_k + \alpha d_k, y_k + \alpha p_k) - AL_k - m_k^{AL}(\alpha)| \\ & \leq M_{AL} \alpha^2 \|d_k\|^2 \end{aligned} \tag{15b}$$

for constants $M_h, M_{AL} > 0$ bounding the second derivatives.

If NST holds and if we assume boundedness of the derivatives of the objective and constraint functions, we also have that there exist constants $M_d, M_p, M_m > 0$, such that $\|d_k\| \leq M_d$, $\|p_k\| \leq M_p$, and $|m_k^{AL}(\alpha)| \leq M_m \alpha$.

B. Filter Line-Search Algorithm

We define a two-dimensional filter that uses the constraint violation and the AL to accept and reject iterates. The filter is initialized at $k = 0$ as

$$\mathcal{F}_0 := \{(h, AL) | h \geq h^{\max}\} \tag{16}$$

with a given parameter $h^{\max} > 0$. We will see that, by choosing AL instead of the objective function $f(\cdot)$ (as is done in standard settings), we can achieve compatibility between the structure of the regularized linear system (6) and the filter globalization procedure.

Given a step (d_k, p_k), a line search is started from counter $\ell \leftarrow 0$ and $\alpha_{k,\ell} = \alpha_k^{\max} \leq 1$ to define trial iterates $x_k(\alpha_{k,\ell}) := x_k + \alpha_{k,\ell} d_k$ and $y_k(\alpha_{k,\ell}) := y_k + \alpha_{k,\ell}\, p_k$. We consider the following conditions to check whether a trial iterate should be accepted.

1) *Filter Condition FC:*

$$(h(x_k(\alpha_{k,\ell})), AL(x_k(\alpha_{k,\ell}), y_k(\alpha_{k,\ell}))) \notin \mathcal{F}_k. \tag{17}$$

2) *Switching Condition SC:*

$$-m_k^{AL}(\alpha_{k,\ell}) > 0 \tag{18}$$

and

$$[-m_k^{AL}(\alpha_{k,\ell})]^{s_{AL}} [\alpha_{k,\ell}]^{1-s_{AL}} > \kappa_h [h(x_k)]^{s_h}. \tag{19}$$

3) *Armijo Condition AC:*

$$AL(x_k(\alpha_{k,\ell}), y_k(\alpha_{k,\ell})) \leq AL_k + \eta_{AL} m_k^{AL}(\alpha_{k,\ell}). \tag{20}$$

4) *Sufficient Decrease Condition SDC:*

$$h(x_k(\alpha_{k,\ell})) \leq h_k - \gamma_h h_k \tag{21}$$

or

$$AL(x_k(\alpha_{k,\ell}), y_k(\alpha_{k,\ell})) \leq AL_k - \gamma_{AL} h_k. \tag{22}$$

Here, $\kappa_h > 0, s_h > 1, s_{AL} = 1, \gamma_{AL}, \gamma_h \in (0,1)$, and $\eta_{AL} \in (0,1)$ are given constants. The filter condition FC is the first requirement for accepting a trial iterate $x_k(\alpha_{k,\ell})$. If the pair $(h(x_k(\alpha_{k,\ell})), AL(x_k(\alpha_{k,\ell}))) \in \mathcal{F}_k$ (i.e., the trial iterate is contained in the filter) the step is rejected and we decrease the step size. If the trial iterate is not contained in the filter, we test for additional conditions. We have the following two possible cases.

1) If SC holds, the step (d_k, p_k) is a descent direction for the AL, and we check whether AC holds. If AC holds, we accept $x_k(\alpha_{k,\ell})$. If not, we decrease the step size.
2) If SC does not hold but SDC does, we accept $(x_k(\alpha_{k,\ell}), y_k(\alpha_{k,\ell}))$. If not, we decrease the step size.

If the trial iterate $(x_k(\alpha_{k,\ell}), y_k(\alpha_{k,\ell}))$ is accepted under SDC, the filter is *augmented* as

$$\mathcal{F}_{k+1} \leftarrow \\ \mathcal{F}_k \cup \{(h, AL) | AL \geq AL_k - \gamma_{AL} h_k, h \geq h_k - \gamma_h h_k\} \tag{23}$$

otherwise, we leave the filter unchanged (i.e., $\mathcal{F}_{k+1} \leftarrow \mathcal{F}_k$). If the trial step size $\alpha_{k,\ell}$ becomes smaller than a threshold $\alpha_k^{\min}$ and the step is not been accepted in either case, we revert to feasibility restoration (find a sufficiently feasible point that is acceptable to the filter) and the filter is augmented. We establish a value for $\alpha_k^{\min}$ in Section III-A. We define the set of iteration counters in which the filter is augmented as $\mathcal{A}$ and the set of counters in which the feasibility restoration phase is denoted as $\mathcal{R}$.

We refer to the first condition of SDC as SDCh to emphasize that this condition accepts the trial iterate if it improves the constraint violation. Similarly, we refer to the second condition as SDCAL to emphasize that this condition accepts the trial iterate if it improves the AL. We refer to successful iterates in which the filter is not augmented (iterates in which SC and AC hold) as *L-iterates*. The filter line-search algorithm is summarized below.

Filter Line-Search Algorithm

1) **Given** starting point x_0, constants $h_{\max} \in (h(x_0), \infty]$, γ_h, $\gamma_{AL} \in (0,1)$, $\eta_{AL} \in (0,1)$, $\kappa_h > 0$, $s_h > 1$, $s_{AL} \geq 1$, $\eta_{AL} \in (0,1)$, and $0 < \tau_2 \leq \tau_1 < 1$.
2) **Initialize** filter $\mathcal{F}_0 := \{(h, AL) : h \geq h_{\max}\}$ and iteration counter $k \leftarrow 0$.
3) **Check Convergence.** Stop if (x_k, y_k) is stationary.
4) **Compute Search Direction.** Compute step (d_k, p_k) that satisfies NST, CT (9), and DPT (8) (with $1/\delta_k^c = 0$ if $h_k = 0$).
5) **Backtracking Line-Search.**
    a) **Initialize**. Set $\alpha_{k,0} \leftarrow \alpha_k^{\max}$ and counter $\ell \leftarrow 0$.
    b) **Compute Trial Point**. If $\alpha_{k,\ell} \leq \alpha_k^{\min}$ revert to feasibility restoration in Step 8. Otherwise, set trial point $x_k(\alpha_{k,\ell}) \leftarrow x_k + \alpha_{k,\ell} d_k$, $y_k(\alpha_{k,\ell}) \leftarrow y_k + \alpha_{k,\ell} p_k$.
    c) **Check Acceptability to the Filter**. If FC (17) does not hold, reject $x_k(\alpha_{k,\ell}), y_k(\alpha_{k,\ell})$, and go to Step 4e.
    d) **Check Sufficient Progress**.
        i) If SC (18) and AC (20) hold, accept $x_k(\alpha_{k,\ell}), y_k(\alpha_{k,\ell})$ and go to Step 5.
        ii) If SC (18) does not hold and SDC (21) hold, accept $x_k(\alpha_{k,\ell}), y_k(\alpha_{k,\ell})$, and go to Step 5. Otherwise, go to Step 4.5.
    e) **New Step Size**. Choose $\alpha_{k,\ell+1} \in [\tau_1 \alpha_{k,\ell}, \tau_2 \alpha_{k,\ell}]$, set $\ell \leftarrow \ell + 1$, and go to Step 4b.
6) **Accept Trial Point**. Set $\alpha_k \leftarrow \alpha_{k,\ell}$, $x_{k+1} \leftarrow x_k(\alpha_{k,\ell})$, and $y_{k+1} \leftarrow y_k(\alpha_{k,\ell})$.
7) **Augment Filter**. If SC (18) is not satisfied, augment filter using (23). Otherwise, leave filter unchanged.
8) **Next Iteration**. Increase iteration counter $k \leftarrow k + 1$ and go to Step 3.
9) **Feasibility Restoration**. Compute an iterate x_{k+1}, y_{k+1} that satisfies FC (17) and SDC (21) and add k to $\mathcal{R}$. Augment filter using (23) (add k to $\mathcal{A}$), and go to Step 7.

Remark: The feasibility restoration phase seeks to find a point that is feasible if the step size becomes smaller than a certain threshold. For more details, the reader is referred to [21].

III. Proof of Global Convergence

Having established basic results and stated the algorithm, we proceed by establishing conditions that guarantee global convergence. We first show that the different step tests proposed yield descent directions for the constraint violation and for the AL. We then show that these properties are sufficient to guarantee that the filter line-search algorithm delivers at least one limit point that is feasible and stationary.

A. Step Computation

We first establish conditions under which the Newton step computed from (6) will be productive in the sense that it will satisfy either SC or SDC. We also prove that the norm of the primal step can be used as a criticality measure.

In a standard filter setting it is possible to prove that, if the KKT system (6) is solved for $\frac{1}{\delta_k^c} = 0$ (dual regularization is not used), then $m_k^h(\alpha) < 0$. If, in addition, h_k is sufficiently small and the reduced Hessian $Z_k^T(W_k + \delta_k^w I)Z_k$ is positive definite, then $m_k^{AL}(\alpha) < 0$ [12, Lemma 3]. In the presence of dual regularization (when $\frac{1}{\delta_k^c} > 0$) or when the reduced Hessian is not positive definite, however, these basic results do not hold. We thus extend existing results to consider these cases. We start by establishing conditions for descent of $h(x)$, which is required to satisfy SDC.

Lemma III.1: If $h(x_k) > 0$, then $m_k^h(\alpha) < 0$.

Proof: From (13) we have that

$$\begin{aligned} m_k^h(\alpha) &= \alpha c_k^T A_k d_k \\ &= \alpha\left(\frac{1}{\delta_k^c} c_k^T p_k - c_k^T c_k\right) \\ &\le \alpha\left(\frac{1}{\delta_k^c}\|c_k\|\|p_k\| - \|c_k\|\|c_k\|\right) \\ &\le 2\alpha(\beta - 1)h_k. \end{aligned}$$

The second equality follows from (7) and the last inequality follows from (3) and DPT (8) with $\beta \in [0, 1)$. ■

Remark: We note that Lemma III.1 holds at *feasible* iterates with $h_k = 0$ (which implies $c_k = 0$ and that we use $1/\delta_k^c = 0$ in the KKT matrix). This is because $A_k d_k = -c_k$ holds.

We now prove that the norm of the primal step can be used as a criticality measure (12).

Theorem III.1: Consider a subsequence $\{x_{k_i}\}$ satisfying $\lim_{i\to\infty} x_{k_i} = x^*$ with $c(x^*) = 0$. Then, $\lim_{i\to\infty} \chi_{k_i} = 0$ implies

$$\lim_{i\to\infty} \|\nabla L_{k_i}\| = 0 \text{ and } \lim_{i\to\infty} \|\nabla AL_{k_i}\| = 0. \tag{24}$$

Proof: Because the subsequence $\{x_{k_i}\}$ converges to a feasible point x^* we have that $\lim_{i\to\infty}\|c_{k_i}\| = 0$. If $\frac{1}{\delta_k^c} > 0$, then S_{k_i} exists and is nonsingular (as a result of NST). From (10), we have that $\|d_{k_i}\| = O(\|\nabla L_{k_i}\|) + O(\|c_{k_i}\|)$. If $\frac{1}{\delta_k^c} = 0$ and because NST holds, A_k must have full row rank and the reduced Hessian $Z_{k_i}^T(W_{k_i} + \delta_k^w I)Z_{k_i}$ for Z_{k_i} spanning the null-space of A_{k_i} must be nonsingular (see [23, Lemma 3.1]). Moreover, we have that $\|d_{k_i}\| = O(\|Z_{k_i}^T g_{k_i}\|) + O(\|c_{k_i}\|)$ and $O(\|Z_{k_i}^T g_{k_i}\|) = O(\|Z_{k_i}^T \nabla L_{k_i}\|) = O(\|\nabla L_{k_i}\|)$ because $Z_{k_i}^T g_{k_i} = Z_{k_i}^T \nabla L_{k_i}$. This guarantees that $\lim_{i\to\infty}\chi_{k_i} = 0$ implies $\lim_{i\to\infty}\|\nabla L_{k_i}\| = 0$ for $\lim_{i\to\infty}\|c_{k_i}\| = 0$, as desired. Because $0 \le \|\nabla AL_{k_i}\| \le \|\nabla L_{k_i}\| + \|\delta^c \nabla c_{k_i}^T c_{k_i}\|$, we have $0 \le \lim_{i\to\infty}\|\nabla AL_{k_i}\| \le \lim_{i\to\infty}\|\nabla L_{k_i}\| + \lim_{i\to\infty}\|\delta^c \nabla c_{k_i}^T c_{k_i}\| = 0$, which completes the proof. ■

We can now state the following result, which states that the Newton step provides a descent direction for the AL. This result is required to satisfy SC.

Lemma III.2: Consider a subsequence $\{x_{k_i}\}$ with $\chi_{k_i} \ge \epsilon$ for $\epsilon > 0$ independent of i and $\delta_{k_i}^c > 0$. Then, there exist constants $\epsilon_1, \epsilon_2 > 0$, such that for $\alpha \in (0, 1]$

$$h_{k_i} \le \frac{1}{2}\epsilon_1^2 \Rightarrow m_{k_i}^{AL}(\alpha) \le -\alpha\epsilon_2.$$

Proof: For simplicity in the notation, we use k instead of k_i and we define $\bar{W}_k := W_k + \delta_k^w I$. Because $\delta_k^c > 0$ holds, from (6), we have that $A_k d_k = \frac{1}{\delta_k^c} p_k - c_k$ holds. In addition, by recalling that $\nabla AL_k = \nabla L_k + \delta^c A_k^T c_k$ we can establish that

$$\begin{aligned} d_k^T \nabla AL_k &= d_k^T(\nabla L_k + \delta^c A_k^T c_k) \\ &= -d_k^T \bar{W}_k d_k - d_k^T A_k^T p_k + \delta^c d_k^T A_k^T c_k \\ &= -d_k^T \bar{W}_k d_k - (p_k^T - \delta^c c_k^T)\left(\frac{1}{\delta_k^c} p_k - c_k\right). \end{aligned}$$

We also have that

$$\begin{aligned} p_k^T(c_k + \alpha A_k d_k) &= p_k^T(c_k - \alpha c_k + \alpha c_k + \alpha A_k d_k) \\ &= (1-\alpha)p_k^T c_k + \alpha p_k^T(c_k + A_k d_k) \\ &= (1-\alpha)p_k^T c_k + \alpha\frac{1}{\delta_k^c} p_k^T p_k. \end{aligned} \tag{25}$$

Combining terms we obtain

$$\begin{aligned} m_k^{AL}(\alpha)/\alpha &= d_k^T \nabla AL_k + p_k^T(c_k + \alpha A_k d_k) \\ &= -d_k^T \bar{W}_k d_k - (1-\alpha)\frac{1}{\delta_k^c} p_k^T p_k \\ &\quad + \left(2 + \frac{\delta^c}{\delta_k^c} - \alpha\right) p_k^T c_k - \delta^c c_k^T c_k. \end{aligned} \tag{26}$$

We know that $\alpha \le 1$ and $\|d_k\| = \chi_k \ge \epsilon$. Moreover, because NST holds, we have that $\|p_k\| \le M_p$. Consequently,

$$\begin{aligned} m_k^{AL}(\alpha)/\alpha &\le -\kappa\epsilon^2 - (1-\alpha)\frac{1}{\delta_k^c} p_k^T p_k \\ &\quad - \delta^c c_k^T c_k + \left(2 + \frac{\delta^c}{\delta_k^c} - \alpha\right) M_p \epsilon_1 \\ &\le -\kappa\epsilon^2 + \left(2 + \frac{\delta^c}{\delta_k^c} - \alpha\right) M_p \epsilon_1. \end{aligned} \tag{27}$$

The result follows with

$$\epsilon_1 = \frac{\kappa\epsilon^2}{2(2 + \frac{\delta^c}{\delta_k^c} - \alpha)M_p} > 0, \quad \epsilon_2 = \frac{1}{2}\kappa\epsilon^2 > 0. \tag{28}$$

This completes the proof. ■

Remark: Note that Lemma III.2 holds for any $\delta_k^c > 0$ and $\delta^c > 0$. From (25) we also have that

$$d_k^T \nabla AL_k \le -\kappa\epsilon^2 + \left(1 + \frac{\delta^c}{\delta_k^c}\right) M_p \epsilon_1.$$

Therefore, if ϵ_1 also satisfies

$$\epsilon_1 = \min\left(\frac{\kappa\epsilon^2}{2(1+\frac{\delta^c}{\delta_k^c})M_p}, \frac{\kappa\epsilon^2}{2(2+\frac{\delta^c}{\delta_k^c}-\alpha)M_p}\right) \tag{29}$$

we have that $d_k^T\nabla AL_k \leq -\alpha\epsilon_2$ and $m_k^{AL}(\alpha) \leq -\alpha\epsilon_2$ hold.

Remark: From (10) we note that the following holds:

$$\begin{aligned} d_k^T S_k d_k &= -d_k^T \nabla L_k - \delta_k^c A_k^T c_k \\ &= -d_k^T \nabla L_k - \delta^c A_k^T c_k + (\delta_c - \delta_k^c) A_k^T c_k \\ &= -d_k^T \nabla AL_k + (\delta_c - \delta_k^c) A_k^T c_k. \end{aligned} \tag{30}$$

We thus have that $d_k^T S_k d_k > 0$ implies that $d_k^T \nabla AL_k < 0$ for sufficiently small h_k. Consequently, when $\delta_k^c > 0$, we can also enforce descent by replacing the CT (9) with a test of the form $d_k^T S_k d_k \geq \kappa d_k^T d_k$ or by enforcing that S_k is positive definite (this can be detected by using a Cholesky factorization).

Remark: Lemma III.2 holds at any feasible iterate with $h_k = 0$ (which implies $c_k = 0$ and that we set $1/\delta_k^c = 0$ in the KKT matrix). This is because we have that $d_k^T \nabla AL_k = -d_k^T \bar{W}_k d_k - d_k^T A_k^T p_k$ and $-d_k^T A_k^T p_k = c_k^T p_k = 0$.

Lemma III.1 guarantees that a step that satisfies DPT yields $h(x_k(\alpha_{k,\ell})) < h_k$ for sufficiently small $\alpha_{k,\ell}$. It is not guaranteed, however, that a trial step size can be obtained that satisfies the sufficient decrease condition (SDC). If SDC does not hold for a sufficiently small step size, a feasibility restoration phase will be needed to obtain a point that is admissible to the filter. We now determine a minimum step size $\alpha_k^{\min}$ that can be used to indicate when to switch to feasibility restoration. To establish this, we first note that m_k^{AL} from (13) is a quadratic model of α of the form:

$$m_k^{AL}(\alpha) := a_k\alpha^2 + b_k\alpha \tag{31}$$

where $a_k := p_k^T A_k d_k$ and $b_k := g_k^T d_k + y_k^T A_k d_k + p_k^T c_k + \delta^c c_k^T A_k d_k$. If a step satisfies $m_k^{AL} < 0$, then from the second condition of SC, we can see that we do not want to revert to restoration for α satisfying

$$[-m_k^{AL}(\alpha)]^{s_{AL}}\alpha^{1-s_{AL}} > \kappa_h h_k^{s_h}. \tag{32}$$

Let $c_k := \kappa_h h_k^{s_h}$, and we note that we have $s_{AL} = 1, m_k^{AL}(0) = 0$, and $c_k \geq 0$. The above condition becomes $-m_k^{AL}(\alpha) > \kappa_h h_k^{s_h}$ and we do not switch to restoration while

$$\begin{cases} \alpha_k \in \left(\frac{-b_k-\sqrt{b_k^2-4a_kc_k}}{2a_k}, \frac{-b_k+\sqrt{b_k^2-4a_kc_k}}{2a_k}\right) \\ \qquad\qquad \text{If } a_k > 0 \\ \alpha_k \in \left(\frac{-c_k}{b_k}, +\infty\right) \qquad \text{If } a_k = 0 \\ \alpha_k \in \left(\frac{-b_k-\sqrt{b_k^2-4a_kc_k}}{2a_k}, +\infty\right) \qquad \text{If } a_k < 0 \end{cases} \tag{33}$$

with $b_k^2 - 4a_kc_k \geq 0$ for all three conditions.

If SC is not satisfied, then the decision to switch to restoration is based on SDC. From Lemma III.1 we have that, if DPT holds, then $m_k^h(\alpha) \leq 2\alpha(\beta-1)h_k$. Consequently, SDCh will hold if $2\alpha(\beta-1)h_k \leq -\gamma_h h_k$ which implies that we do not want to revert to restoration if

$$\alpha \geq \frac{\gamma_h}{2-2\beta}. \tag{34}$$

Note that the standard (unregularized) filter algorithm uses $\frac{1}{\delta_k^c} = 0$ and we thus have that the above condition reduces to $\alpha \geq \frac{1}{2}\gamma_h$. The last case to consider is that in which SC is not satisfied due to its second condition SDCAL. Condition SDCAL states that we do not want to revert to restoration if

$$m_k^{AL}(\alpha) \leq -\gamma_{AL} h_k. \tag{35}$$

We define $\hat{c}_k := \gamma_{AL} h_k$ and we do not revert to restoration if

$$\begin{cases} \alpha_k \in \left[\frac{-b_k-\sqrt{b_k^2-4a_k\hat{c}_k}}{2a_k}, \frac{-b_k+\sqrt{b_k^2-4a_k\hat{c}_k}}{2a_k}\right] \\ \qquad\qquad \text{If } a_k > 0 \\ \alpha_k \in \left[\frac{-\hat{c}_k}{b_k}, +\infty\right) \qquad \text{If } a_k = 0 \\ \alpha_k \in \left[\frac{-b_k-\sqrt{b_k^2-4a_k\hat{c}_k}}{2a_k}, +\infty\right) \qquad \text{If } a_k < 0 \end{cases} \tag{36}$$

$b_k^2 - 4a_k\hat{c}_k \geq 0$ for all three conditions.

Considering (33), (34), and (36) altogether, we have that the minimum step size triggering restoration is

$$\alpha_k^{\min} := \begin{cases} \text{If } m_k^{AL} < 0 \text{ and } a_k = 0 \\ \qquad \gamma_\alpha \min\left\{\frac{\gamma_h}{2-2\beta}, \frac{-c_k}{b_k}, \frac{-\hat{c}_k}{b_k}\right\} \\ \text{If } m_k^{AL} < 0, a_k > 0 \\ \qquad \gamma_\alpha \min\left\{\frac{\gamma_h}{2-2\beta}, \frac{-b_k-\sqrt{b_k^2-4a_k\min(c_k,\hat{c}_k)}}{2a_k}\right\} \\ \text{otherwise} \\ \qquad \gamma_\alpha \frac{\gamma_h}{2-2\beta} \end{cases} \tag{37}$$

where $b_k^2 - 4a_k\min(c_k, \hat{c}_k) \geq 0$ holds for all the three conditions and $\gamma_\alpha \in (0,1)$ is a safeguard parameter.

B. Feasibility

We now prove that the algorithm eventually delivers a subsequence of iterates that yield a feasible solution.

Lemma III.3: At all feasible but not optimal iterates k (i.e., $h_k = 0$ and $\chi_k > 0$) we have that

$$\Phi_k := \min\{h : (h, AL) \in \mathcal{F}_k\} > 0$$

for all $\alpha \in (0, 1]$.

Proof: Observe that $\Phi_k > 0$ for $k = 0$. Suppose now that the claim is true for $k-1$. If $h_{k-1} > 0$ and d_{k-1} is such that the filter is augmented in iteration $k-1$ by (23), then we have $\Phi_k > 0$ because $\gamma_h \in (0,1)$. From Lemma III.1 we know that such a step d_k exists. If, on the other hand, $h_{k-1} = 0$, then we know from Lemma III.2 that $m_{k-1}^{AL}(\alpha) < 0$ for all $\alpha \in (0,1]$ and the switching condition SC must hold. In addition, from (37) we have $\alpha_k^{\min} = 0$, and consequently the back-tracking procedure is eventually able to find an acceptable step size $\alpha \geq \alpha_k^{\min}$ that satisfies AC. Therefore, the filter is not augmented in this case and $\Phi_k = \Phi_{k-1} > 0$. ■

The above result states that the filter never includes feasible points (i.e., satisfying $h_k = 0$).

Lemma III.4: If the filter is only augmented a finite number of times (i.e., $|\mathcal{A}| < \infty$), then

$$\lim_{k\to\infty} h_k = 0. \tag{38}$$

Proof: We choose an iteration K such that for all $k \geq K$ the filter is not augmented. We thus have that SC and AC hold for $k \geq K$. From [12, Lemma 5] we know that there exists $c_4 > 0$ such that for $k \geq K$

$$AL_{k+1} - AL_k < -c_4 h_k^{s_h}. \tag{39}$$

Consequently, for all $i = 1, 2, \dots,$

$$AL_{K+i} < AL_K - c_4 \sum_{k=K}^{k+i-1} h_k^{s_h}. \tag{40}$$

Because $AL(\cdot)$ is bounded from below as $i \to \infty$, the series $\sum_{k=K}^{k+i-1} h_k^{s_h}$ is bounded from above which implies that $\lim_{i\to\infty} \sum_{k=K}^{k+i-1} h_k^{s_h} = 0$. The result follows. ■

Lemma III.5: Consider a subsequence $\{x_{k_i}\}$ such that the filter is augmented at iteration k_i. Then

$$\lim_{i\to\infty} h_{k_i} = 0. \tag{41}$$

Proof: Since we have continuity and boundedness of the objective and constraints, we know that there exist constants c_L and c_h such that $AL_{k_i} \geq c_{AL}$ and $h_{k_i} \leq c_h$ for all i. Lemma III.1 guarantees that steps d_{k_i} satisfying SDC (either SDCh or SDCAL) exist and therefore satisfy FC. The result follows from [23, Lemma 3.3]. ■

Theorem III.2: The algorithm delivers a sequence of iterates $\{x_k\}$ satisfying

$$\lim_{k\to\infty} h_k = 0. \tag{42}$$

Proof: The NST condition guarantees the existence of search steps. DPT guarantees that iterates $k \in \mathcal{A}$ exist that provide sufficient reduction in the constraint violation. Moreover, the limit exists because $h_k = 0$ implies $1/\delta_k^c = 0$. In addition, CT guarantees that iterates $k \notin \mathcal{A}$ exist. The proof thus follows along the lines of [12, Theorem 1]. ■

Remark: At this point it is important to highlight that the algorithm provides some flexibility to handle rank-deficient Jacobians but only at nonfeasible points (i.e., $h_k > 0$). In particular, even if the Jacobian is rank deficient, we can choose δ_k^w and δ_k^c appropriately to satisfy NST, DPT, and CT. However, since we require that $1/\delta_k^c = 0$ whenever $h_k = 0$, the NST condition can only be satisfied when the Jacobian has full row rank (particularly at a stationary solution). Designing algorithms that converge to points with rank-deficient Jacobians is significantly more challenging and beyond the scope of this paper (for approaches dealing with such problems, the reader is referred to [24]). We highlight, however, that one of the main objectives behind the design of the proposed algorithm is to allow for primal–dual regularization of the KKT system (even if the NLP has full rank Jacobians) in order to avoid solving a KKT system in the primal–dual space. In this respect, the algorithm provides great computational flexibility.

C. Optimality

We now prove that the algorithm eventually delivers a subsequence of iterates that yield a feasible and optimal (stationary) solution for the NLP.

Lemma III.6: Let $\{x_{k_i}\}$ be a subsequence satisfying $m_{k_i}^{AL} \leq -\alpha\epsilon_2$ for $\epsilon_2 > 0$ independent of k_i and for all $\alpha \in (0, 1]$. There exists a step size threshold $\bar{\alpha} > 0$, such that for all k_i and $\alpha \leq \bar{\alpha}$ we have that

$$AL(x_{k_i} + \alpha d_{k_i}, y_{k_i} + \alpha p_{k_i}) - AL(x_{k_i}, y_{k_i}) \leq \gamma_{AL} m_k^{AL}(\alpha). \tag{43}$$

Proof: From (15) and because CT guarantees that there exists (d_{k_i}, p_{k_i}) such that $m_{k_i}^{AL}(\alpha) < 0$ we know that

$$\begin{aligned} &AL(x_{k_i} + \alpha d_{k_i}, y_{k_i} + \alpha p_{k_i}) - AL_{k_i} - m_{k_i}^{AL}(\alpha) \\ \leq\ & M_{AL}\alpha^2 \|d_{k_i}\|^2. \end{aligned}$$

Because NST holds, we have that $\|d_{k_i}\| \leq M_d$ and for $\alpha \leq \bar{\alpha} := \frac{(1-\gamma_{AL})\epsilon_2}{M_{AL} M_d^2}$ we have that

$$\begin{aligned} &AL(x_{k_i} + \alpha d_{k_i}, y_{k_i} + \alpha p_{k_i}) - AL_{k_i} - m_{k_i}^{AL}(\alpha) \\ \leq\ & -(1-\gamma_{AL}) m_{k_i}^{AL}(\alpha). \end{aligned}$$

The result follows. ■

Lemma III.7: Assume the filter is augmented only a finite number of times (i.e., $|\mathcal{A}| < \infty$). Then

$$\lim_{k\to\infty} \chi_k = 0.$$

Proof: Because $|\mathcal{A}| < \infty$, there exists K so that $k \notin \mathcal{A}$ for $k \geq K$. Suppose the claim is not true and there exists a subsequence $\{x_{k_i}\}$ for which $\chi_{k_i} \geq \epsilon$ with $\epsilon > 0$. Because CT holds we have from Lemma III.4 that $\lim_{i\to\infty} h_{k_i} = 0$, because the filter is augmented only a finite number of times. Consequently, there exists $\bar{K} \geq K$ and $\epsilon_1 > 0$ such that $h_{k_i} \leq \frac{1}{2}\epsilon_1$ for all $k_i \geq \bar{K}$. From Lemma III.2 we know that there exists $\epsilon_2 > 0$ such that for all $k_i \geq \bar{K}$ we have that $m_{k_i}^{AL}(\alpha) \leq -\alpha\epsilon_2$ for all $\alpha \in (0, 1]$. Because the filter is not augmented in k_i, we have that the AC is eventually satisfied:

$$AL(x_{k_i} + \alpha_{k_i} d_{k_i}, y_{k_i} + \alpha_{k_i} p_{k_i}) - AL_{k_i} \leq -\alpha_{k_i}\eta_{AL}\epsilon_2. \tag{44}$$

Following the reasoning of Lemma III.4 we have that AL_{k_i} is monotonically decreasing and because it is bounded below we have that $\lim_{i\to\infty} \alpha_{k_i} = 0$. Assume now that $\bar{K}$ is large enough so that $\alpha_{k_i} < 1$ which means that for $k_i \geq \bar{K}$ the trial step size $\alpha_{k,0} = 1$ was not accepted. The last trial step $\alpha_{k_i,\ell_i} \geq \alpha_{k_i}$ must have satisfied SC (by construction). Therefore, the last trial step must have been rejected because AC does not hold and thus

$$\begin{aligned} &AL(x_{k_i} + \alpha_{k_i,\ell_i} d_{k_i}, y_{k_i} + \alpha_{k_i,\ell_i} p_{k_i}) - AL_{k_i} \\ &\quad \geq -\alpha_{k_i,\ell_i} m_{k_i}^{L}(\alpha_{k_i,\ell_i}) \end{aligned} \tag{45}$$

or, alternatively, it must have been rejected due to the fact that it is in filter and thus

$$\begin{aligned} &(h(x_{k_i} + \alpha_{k_i,\ell_i} d_{k_i}), AL(x_{k_i} + \alpha_{k_i,\ell_i} d_{k_i}, y_{k_i} + \alpha_{k_i,\ell_i} p_{k_i})) \\ &\quad \in \mathcal{F}_{k_i} = \mathcal{F}_K. \end{aligned} \tag{46}$$

We now argue that neither of these two situations can occur. First, AC must always hold because $\lim_{i\to\infty} \alpha_{k_i} = \lim_{i\to\infty} \alpha_{k_i,\ell_i} = 0$ implies that there exists $\bar{\alpha} > 0$ such that $\alpha_{k_i,\ell_i} \le \bar{\alpha}$ and from Lemma III.6 we have established that (45) does not hold. Second, if we let $\Phi_K := \min\{h : (h, AL) \in \mathcal{F}_K\}$, then from Lemma III.3 we have that $\Phi_K > 0$. If DPT holds, we have from (15) and Lemma III.1 that there exists α_{k_i,ℓ_i} such that

$$\begin{aligned} & h(x_{k_i} + \alpha_{k_i,\ell_i} d_{k_i}) \\ & \quad \le h_{k_i} - 2\alpha_{k_i,\ell_i}(1-\beta)h_{k_i} + M_h M_d^2 \alpha_{k_i,\ell_i}^2 \end{aligned} \tag{47}$$

for $\beta \in [0, 1)$. Because $\lim_{i\to\infty} \alpha_{k_i,\ell_i} = 0$ and $\lim_{i\to\infty} h_{k_i} = 0$ we have from Lemma III.4 that, for k_i large enough, $h(x_{k_i} + \alpha_{k_i,\ell_i} d_{k_i}) < \Phi_K$ holds, which is a contradiction to (46). ■

The next lemma is needed to guarantee that, under the step computation mechanism, it is possible to find a step size that is acceptable to the filter.

Lemma III.8: Let $\{x_{k_i}\}$ be a subsequence with $k_i \notin R_{\text{inc}}$ and $m_{k_i}(\alpha) \le -\alpha\epsilon_2$ for $\epsilon_2 > 0$ independent of k_i and for all $\alpha \in (0, 1]$. There exist constants $c_5, c_6 > 0$ such that

$$(h(x_{k_i} + \alpha d_{k_i}), AL(x_{k_i} + \alpha d_{k_i}, y_{k_i} + \alpha p_{k_i})) \notin \mathcal{F}_{k_i} \tag{48}$$

for all k_i and $\alpha \le \min\{c_5, c_6 h_{k_i}\}$.

Proof: By construction we know that a current iterate x_{k_i} is not in the filter (i.e., $(h_{k_i}, AL_{k_i}) \notin \mathcal{F}_{k_i}$). Furthermore, we know that this implies that $(h, AL) \notin \mathcal{F}_{k_i}$ if $AL \le AL_{k_i}$ and $h \le h_{k_i}$ and therefore that an iterate satisfying such conditions is acceptable to the filter. If we let $c_6 := \frac{2(1-\beta)}{M_h M_d^2}$, then for $\alpha \le c_6 h_{k_i} = \frac{2(1-\beta)h_{k_i}}{M_h M_d^2}$ we have that

$$\begin{aligned} m_{k_i}^h(\alpha) + M_h \alpha^2 M_d^2 & \le -2\alpha(1-\beta)h_{k_i} + M_h \alpha^2 M_d^2 \\ & \le 0 \end{aligned} \tag{49}$$

where the first inequality follows from Lemma III.1. Therefore, from the Taylor bounds (15), we have that $h(x_{k_i} + \alpha d_{k_i}) - h_{k_i} \le m_{k_i}^h(\alpha) + M_h \alpha^2 M_d^2$, which results in

$$h(x_{k_i} + \alpha d_{k_i}) \le h_{k_i}. \tag{50}$$

Similarly, if CT holds, we have from Lemma III.2 that there exists $m_{k_i}^{AL}(\alpha) \le -\alpha\epsilon_2$ (at least for sufficiently small h_{k_i}). In this case, we have that

$$AL(x_{k_i} + \alpha d_{k_i}, y_{k_i} + \alpha p_{k_i}) \le AL_{k_i} \tag{51}$$

for $\alpha \le c_5$ and $c_5 := \min\{1, \epsilon_2/(M_d^2 M_{AL})\}$. ■

The following lemma corresponds to [12, Lemma 10] and proves that for a subsequence of nonoptimal iterates *the filter is eventually not augmented*. We only state the result for completeness. The proof does not change.

Lemma III.9: Let $\{x_{k_i}\}$ be a subsequence with $\chi_{k_i} \ge \epsilon$ for $\epsilon > 0$ independent of k_i. There exists K such that for all $k_i \ge K$ the filter is not augmented (i.e., $k_i \notin \mathcal{A}$).

We are now in a position to state the main convergence result.

Theorem III.3: The algorithm delivers a sequence of iterates $\{x_k\}$ satisfying:

$$\lim_{k\to\infty} h_k = 0 \qquad \text{and} \qquad \liminf_{k\to\infty} \chi_k = 0.$$

Proof: The first result follows from Theorem III.2. To establish the second result we consider two cases. First, if the filter is augmented only a finite number of times, then Lemma III.7 already establishes the result. Second, consider that there exists a subsequence $\{x_{k_i}\}$ such that $k_i \in \mathcal{A}$ for all i. Now suppose that $\limsup_{i\to\infty} \chi_{k_i} > 0$. Then, by construction of the algorithm, there exists a subsequence $\{x_{k_{i_j}}\}$ of $\{x_{k_i}\}$ such that $\lim_{j\to\infty} h_{k_{i_j}} = 0$ and $\chi_{k_{i_j}} \ge \epsilon$ for all k_{i_j}. Lemma III.9 guarantees that there exists sufficiently small k_{i_j} such that the filter is not augmented and thus $k_{i_j} \notin \mathcal{A}$. This contradicts the assumption on the subsequence $\{x_{k_i}\}$ and thus such a subsequence cannot exist. The result follows. ■

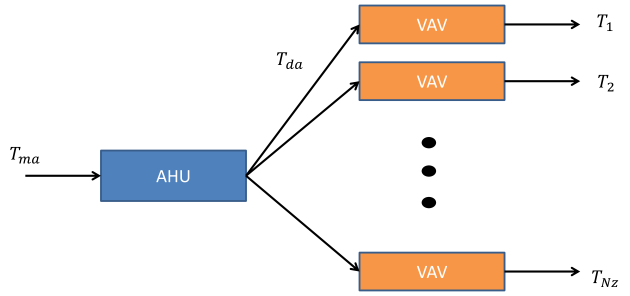


Fig. 1. Schematic representation of HVAC system.

Remark: We can use the proposed approach to address inequality constraints within a logarithmic interior-point framework. In particular, for a problem of the form:

$$\min_x \; f(x) \quad \text{s.t.} \quad c(x) = 0, \; x \ge 0 \tag{52}$$

we can solve a sequence of equality-constrained barrier subproblems

$$\min_x \; \varphi(x) := f(x) - \mu \sum_{i=1}^{n_x} \log x_i \quad \text{s.t.} \quad c(x) = 0 \tag{53}$$

for decreasing values of the barrier parameter $\mu > 0$. The proposed algorithm can thus be applied by using the barrier $\varphi(x)$ instead of the objective function $f(x)$ and by using a primal–dual approximation of the Hessian of $\varphi(x)$. The barrier subproblems problems are warm-started by using the solution of the previous problem. We also note that the barrier subproblems do not have to be solved to optimality in practice. The filter is initialized at the beginning of every barrier subproblem. For a detailed discussion, the reader is referred to [21] and [13].

IV. Numerical Studies

In this section, we present a case study to demonstrate the practical benefits of the proposed algorithmic framework.

A. HVAC System Description

The HVAC optimization model under study consists of an AHU that provides air flows to multiple VAV boxes that in turn regulate the temperature of individual building zones (see Fig. 1). Specifically, the goal of the AHU optimization system is to determine optimal mixed air and discharge temperatures for the AHU as well as optimal supply flow allocations for the VAV boxes that satisfy as many zone thermal load demands as possible. An AHU heating coil model is used to estimate the total amount of thermal power consumption required by the AHU. We summarize the parameters and variables used in the description of the HVAC optimization model as follows.

Parameters:

1) N_z: number of zones.
2) $P_i^{\max}$: cooling/heating capacity (in Watts) of the ith zone.
3) T_i: desired temperature at the ith zone.
4) $q_i^{\min}$, $q_i^{\max}$: bounds of the air flow rate at the ith zone.
5) $T_{da}^{\min}$, $T_{da}^{\max}$: bounds of the discharge air temperature.
6) $T_{na}^{\min}$, $T_{ma}^{\max}$: bounds of the mixed air temperature.
7) c_{air}: air specific heat capacity.
8) ρ: penalty parameter.

Variables:

1) q_i: air flow rate at the ith zone.
2) T_{da}: discharge air temperature of the AHU.
3) T_{ma}: mixed air temperature.
4) s: slack variable, defined as $s = |T_{da} - T_{ma}|$.

The problem formulation is

$$\min_{q_i, T_{da}, T_{ma}, s} \quad - \sum_{i=1}^{N_z} ((c_{\text{air}} q_i (T_{da} - T_i)) - P_i^{\max}) + \rho \cdot \left(c_{\text{air}} \cdot s \cdot \sum_{i=1}^{N_z} q_i \right) \tag{54a}$$

$$\text{s.t.} \quad c_{\text{air}} \cdot q_i \cdot (T_{da} - T_i) - P_i^{\max} \leq 0, \quad i = 1, \ldots, N_z \tag{54b}$$

$$- s \leq T_{da} - T_{ma} \leq s, \tag{54c}$$

$$q_i^{\min} \leq q_i \leq q_i^{\max}, \tag{54d}$$

$$T_{da}^{\min} \leq T_{da} \leq T_{da}^{\max}, \tag{54e}$$

$$T_{ma}^{\min} \leq T_{ma} \leq T_{ma}^{\max}, \tag{54f}$$

$$s \geq 0. \tag{54g}$$

Constraint (54b) requires the heat supply capacity to be bounded by the required capacity. The thermal power required to raise the inlet mixed air temperature T_{ma} to the discharge air temperature T_{da} is given by $q \cdot |T_{da} - T_{ma}|$. Assuming that there are no air leaks associated to the transport of air between the AHU and VAV system, the total AHU flow rate equals the air supplied to all the VAV boxes. The thermal power consumption of each VAV reheat coil unit is determined based on the heat transfer between the inlet and outlet air flow. The power required by the VAV to heat up the inlet air flow at temperature T_{da} (from the outlet of AHU) to reach the required air temperature T_i at the ith zone is given by $c_{\text{air}} \cdot q_i \cdot (T_{da} - T_i)$. The optimization model (54) is a simplified version of the problem presented in [3]. In particular, we do not include other components such as the electrical power of the supply fan and the air recycle system.

B. Software Implementation

We implemented the proposed algorithm in PIPS-NLP, an object-oriented interior-point framework written in C++ that facilitates the development of algorithmic strategies [6]. The proposed algorithm solves a sequence of barrier subproblems. Precautions are taken to ensure that barrier terms and associated derivatives remain bounded, as explained in [5]. Our implementation allows us to compare the following algorithmic strategies.

1) **Aug**: solves the augmented system (6) with default filter line-search setting that uses objective function and constraint violation (3) as filter entries.
2) **AugAL**: solves the augmented system (6) and uses the AL function (4) and constraint violation (3) as filter entries.
3) **SCAL**: solves the Schur system (10) and uses the AL function (4) and constraint violation (3) as filter entries.

TABLE I
SUMMARY OF ALGORITHMIC STRATEGIES

Method	LinSys	Solver	Filter	Dual Reg.
Aug	(6)	MA27	(f, h)	No
AugAL	(6)	MA27	(AL, h)	Yes
SCALD	(10)	LAPACK	(AL, h)	Yes
SCALS	(10)	CHOLMOD	(AL, h)	Yes

For the Aug and AugAL strategies, we compute the search step by solving the linear system (6) using a symmetric indefinite factorization implemented in MA27 [25]. We trigger primal regularization if the inertia reported by MA27 is not correct [5]. For the SCAL strategy, we use a Cholesky factorization to factorize the Schur complement matrix S_k in (11) and to perform the corresponding backsolve (10). We use two different packages to perform the factorization of S_k. We use a dense Cholesky factorization (SCALD) using libraries dpotrf and dpotrs from the LAPACK suite [26] and we use the CHOLMOD package [27] to perform a sparse Cholesky factorization (SCALS). If the Schur complement matrix is not positive definite, we update the primal regularization parameter δ_k^w until S_k becomes positive definite and CT holds. We revert to solve the augmented system with MA27 if we encounter a feasible but nonoptimal iterate ($h_k = 0$ holds at a nonstationary point). Such a case was not observed in our tests, as is quite common in practice. We summarize the strategies in Table I. Parameter β of the DPT test (8) plays a key role in the performance of the algorithm. A small β value requires dual regularization to be sufficiently small but it may create an ill-conditioned Schur system (10) because it will require a large δ_k^c in (11). On the other hand, a large β value will increase the chance to invoke the restoration phase, since $\frac{\gamma_h}{2-2\beta}$ in (37) becomes larger. Currently, we fix $\beta = 0.5$ in our implementation, which gives $\frac{\gamma_h}{2-2\beta} = \gamma_h$ in (37).

We verified the implementation of our proposed algorithm (AugAL) by performing benchmarks against the standard filter line-search implementation of IPOPT (Aug). We performed tests on 287 CUTEr instances (small to medium instances) [28]. The performance profiles are presented in Fig. 2. In summary, AugAL can solve 80% of the instances, while Aug can solve 91%. The failed cases either require feasibility restoration or reach the maximum number iterations allowed. Given that our implementation is nonoptimized (not tuned), we deem this off-the-shelf performance as acceptable. These preliminary

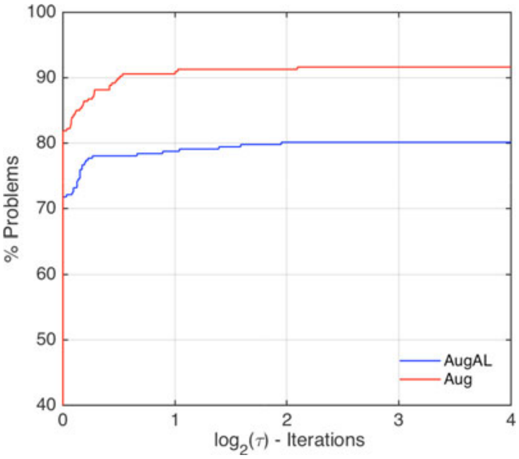


Fig. 2. Performance profiles for Aug and AugAL algorithms.

TABLE II
SUMMARY OF NUMBER OF ITERATIONS FOR SMALL HVAC INSTANCE

$1/\delta_k^c$	δ^c						Primal Reg.
	0	10^{-6}	10^{-2}	10^{+0}	10^{+2}	10^{+6}	
10^{-5}	–	–	–	–	–	55	10
10^{-6}	41	41	41	41	41	41	11
10^{-7}	41	41	41	41	40	40	11
10^{-8}	41	41	41	41	41	41	11
10^{-9}	41	41	41	41	41	41	11
10^{-10}	42	42	42	42	42	42	11
10^{-11}	42	42	42	42	43	43	11
10^{-12}	47	47	47	44	44	44	12
10^{-13}	56	56	56	56	56	56	14
10^{-14}	107	107	107	107	107	107	88
0	49						11

benchmarking results highlight, however, the fact that AugAL is designed as a method to tackle specific problem classes and will likely be restricted in performance compared to a general method such as Aug. In particular, a key tradeoff arises from the fact that primal–dual regularization used in AugAL enables more modular linear algebra and faster embedded implementations (compared to Aug) but this comes at the expense of general performance. In the following results, we present an actual application to highlight these tradeoffs and show that AugAL also provides flexibility to perform algorithmic tuning.

C. Results

We now test the algorithmic strategies by using an HVAC system with 19 different zones. The optimization model has 22 variables, 21 inequality constraints, and 22 inequality constraints. After introducing slack variables into the system, the dimension of the augmented matrix (6) is 64 and its number of nonzeros is 167. The dimension of the Schur system for the AL approach (11) is 43 and it has 146 nonzero entries. We first tested the algorithms on a standard workstation equipped with a 2.8 GHz Dual-Core AMD Opteron(tm) 8220 SE processor to analyze how sensitive the algorithm was to its algorithmic parameters. The workstation has 16 GB of memory available.

In Table II, we present the number of iterations required by the solver with different parameter settings. The last row in the table are the results for the Aug strategy. The rest of the results are for the SCALS strategy with different combinations of δ_k^c and δ^c. We highlight that, in these experiments, δ_k^c is left fixed throughout the iteration history (this allow us to solve the Schur system at each iteration). Keeping δ_k^c fixed also allow us to have a more systematic analysis of algorithmic behavior. In Table II, we also report the number of additional factorizations due to primal regularization required for correcting the positive definiteness of S_k in the SCAL strategies or for correcting the inertia of the KKT matrix in the Aug strategies. From Table II we can see that the proposed SCAL algorithm can solve the test problem for a wide range of values of δ^c and δ_k^c. All solutions achieve the same optimal objective value. The ability to handle a wide range of values of δ_k^c is desirable, as this provides flexibility to improve the conditioning of matrix S_k. Notably, the best parameter setting leads to 40 iterations, and 56 (out of 80) parameter settings achieve less iterations than the standard filter-line search approach Aug. This is because the SCAL strategies provide more flexibility to accept steps than the standard filter line-search approach. We also found that very small values of δ_k^c trigger primal regularization more often. This is because, for such cases, the contribution of the $\delta_k^c A_k A_k^T$ term in S_k is dominated by that of the Hessian matrix. As more primal regularization is added, the search steps are of decreased quality, thus increasing the number of iterations.

We also compare the computational performance of the approaches on an embedded platform. The embedded system we used is a BeagleBoneBlack, which is equipped with a AM335x 1 GHz ARM Cortex-A8 processor. Notably, the embedded system only has 512 MB of memory available. In order to test scalability, we also created a larger example by duplicating the real smaller building system. The large HVAC example contains 1000 variables and 999 inequality constraints. We solve each case study ten times and report averages. Tables III and IV present solution times for the small and large HVAC example, respectively. We use (nIter) to denote the total number of iterations, (nFact) represents the total number of factorizations, and (nSol) represents the total number of backsolves. Fact(μs), BSol(μs) and Schur(μs) represent the total time (in microseconds) required for factorization, backsolve, and construction of the Schur complement matrix (11), respectively. The last row AvgIter(μs) denotes the average time spent on solving the linear system in a single iteration (i.e., the sum of Fact, Sol, and Schur divided by nIter).

By comparing the times obtained for the workstation and embedded platform, it becomes clear that the embedded system is ten times slower than the workstation. This illustrates the limitations of low-cost embedded systems. We also see that the average time per iteration of the standard Aug strategy is drastically reduced by the SCALS counterpart. For the small HVAC system, SCALS improves the performance of Aug by a factor of 3 on a workstation and by a factor of 1.2 on the embedded platform. *Most notably, for the large HVAC example, SCALS improves Aug by a factor of 1437 on a workstation*

TABLE III
PERFORMANCE OF ALGORITHMIC STRATEGIES ON SMALL HVAC PROBLEM

	Desktop				Embedded			
	Aug	AugAL	SCALS	SCALD	Aug	AugAL	SCALS	SCALD
nIter	49	45	42	45	44	45	42	45
nFact	65	61	53	56	59	61	53	56
nSol	129	134	271	214	119	134	271	212
Fact (μs)	4625	4359	645	1054	39 329	30 968	15 636	50 501
Sol (μs)	1073	1102	644	1319	6134	5458	15 740	41 073
Schur (μs)	–	–	264	244	–	–	4689	6146
AveIter (μs)	116	121	36	58	1033	809	858	2171

TABLE IV
PERFORMANCE OF ALGORITHMIC STRATEGIES ON LARGE HVAC PROBLEM

	Desktop				Embedded			
	Aug	AugAL	SCALS	SCALD	Aug	AugAL	SCALS	SCALD
nIter	39	35	35	35	39	35	35	35
nFact	69	65	48	48	69	65	48	48
nSol	133	138	222	166	133	138	222	172
Fact (μs)	84 154 812	70 768 120	25 101	13 385 834	871 159 612	716 593 235	303 852	1 393 419 540
Sol (μs)	680 684	681 306	15 361	3 951 190	4 782 804	4 643 203	229 265	36 876 921
Schur (μs)	–	–	12496	41472	–	–	153 337	163 153
AveIter (μs)	2 175 269	2 041 412	1513	496 528	22 460 061	20 606 755	19 612	40 870 274

and 1145 on the embedded platform. This result illustrates that the proposed approach can significantly increase functionality on embedded platforms compared to off-the-self approaches. We also highlight that the total solution times for the large HVAC system *are on the order of 0.68 s for the SCALS strategy and on the order of 875 s for the Aug strategy*. We attribute these large speed-ups to the ability of the SCALS strategy to operate on a linear system of reduced dimension and to the ability to use linear algebra strategies with reduced overheads. In particular, a Cholesky factorization does not need to perform a symbolic factorization and pivoting, as is done in the indefinite factorization routine implemented in `MA27`.

We note that the number of iterations differs from the number of factorizations and the number of backsolves. This is because of the additional factorizations required for primal regularization and because we perform iterative refinement to achieve a more accurate solution of the linear system. We also observe that approaches Aug and AugAL require more factorizations due to the default setting of `MA27`. In particular, `MA27` reallocates memory if its original allocated memory is not sufficient to accommodate fill-in. This also involves additional factorizations. In particular, for the small example, `MA27` requires 5 additional factorizations while it requires 17 additional factorizations for the large example. This, again, highlights that it is beneficial to use factorization strategies with fixed pivoting such as Cholesky in memory-constrained settings.

We also found that SCALS and SCALD require more iterative refinement steps. This is because SCAL algorithms operate on the reduced Schur system (10) but refine the residual on the entire primal–dual space. We also note that, as discussed in [5], [29], the reduced system will tend to have a larger condition number than the augmented system. Interestingly, even if SCALS and SCALD require significantly more backsolves, we still can observe significant speed up in the factorization step. This is particularly evident in the large HVAC instance.

V. CONCLUSION AND FUTURE WORK

We presented a filter line-search algorithm for nonconvex continuous optimization that uses an AL function and a constraint violation metric to accept and reject steps. The approach is motivated by real-time optimization applications that need to be executed on low-cost embedded computing platforms with constrained memory and processor speeds. The proposed method enables the use of primal–dual regularization of the augmented system that in turn provides flexibility to use linear algebra strategies with lower computing overheads. We prove that the proposed algorithm is globally convergent and can handle negative curvature and nearly rank-deficient Jacobians. We demonstrate the developments by using a set-point optimization application for a building HVAC system. Our numerical studies demonstrate that the proposed approach enables reductions in solution times by a factor of over 1000 on an embedded platform. As part of future work, we are interested in developing strategies to accelerate computations on embedded platforms by using single precision preconditioners and by using field-programmable gate arrays. We are also interested in studying the local asymptotic convergence properties of the proposed algorithm and to explore the possibility of adjusting the AL penalty parameter adaptively.

Acknowledgment

The authors would like to thank V. Adetola, M. Baric, P. Li, and T.E. Lovett for technical discussions on the HVAC model, and S. Leyffer for technical discussions on filter methods.

References

[1] J. L. Jerez, P. J. Goulart, S. Richter, G. A. Constantinides, E. C. Kerrigan, and M. Morari, "Embedded online optimization for model predictive control at megahertz rates," *IEEE Trans. Automat. Control*, vol. 59, no. 12, pp. 3238–3251, Dec. 2014.

[2] V. Adetola *et al.*, "Energy performance monitoring and optimization system for dod campuses," United Technol. Res. Center, East Hartford, CT, USA, Tech. Rep. EW-201142, 2014.

[3] V. Adetola *et al.*, "Model predictive control and fault detection and diagnostics of a building heating, ventilation, and air conditioning system," in *Proc. Int. High Perform. Buildings Conf.*, 2014. [Online]. Available: http://docs.lib.purdue.edu/ihpbc/152

[4] J. Gondzio, "Interior point methods 25 years later," *Eur. J. Oper. Res.*, vol. 218, no. 3, pp. 587–601, 2012.

[5] M. Ulbrich, S. Ulbrich, and L. N. Vicente, "A globally convergent primal-dual interior-point filter method for nonlinear programming," *Math. Program.*, vol. 100, no. 2, pp. 379–410, 2004.

[6] N.-Y. Chiang, C. G. Petra, and V. M. Zavala, "Structured nonconvex optimization of large-scale energy systems using pips-nlp," in *Proc. Power Syst. Comput. Conf.*, 2014, pp. 1–7.

[7] J. Kang, N. Chiang, C. D. Laird, and V. M. Zavala, "Nonlinear programming strategies on high-performance computers," in *Proc. IEEE Conf. Decision Control, Osaka, Japan*, 2015, pp. 4612–4620.

[8] E. Kerrigan, G. Constantinides, A. Suardi, A. Picciau, and B. Khusainov, "Computer architectures to close the loop in real-time optimization," in *Proc. 54th IEEE Conf. Decision Control*, 2015, pp. 4597–4611.

[9] C. V. Rao, S. J. Wright, and J. B. Rawlings, "Application of interior-point methods to model predictive control," *J. Optim. Theory Appl.*, vol. 99, no. 3, pp. 723–757, 1998.

[10] Y. Wang and S. Boyd, "Fast model predictive control using online optimization," *IEEE Trans. Control Syst. Technol.*, vol. 18, no. 2, pp. 267–278, Mar. 2010.

[11] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, "Efficient interior point methods for multistage problems arising in receding horizon control," in *Proc. IEEE 51st Annu. Conf. Decision Control*, 2012, pp. 668–674.

[12] A. Wächter and L. T. Biegler, "Line search filter methods for nonlinear programming: Motivation and global convergence," *SIAM J. Optim.*, vol. 16, no. 1, pp. 1–31, 2005.

[13] N. Chiang and V. M. Zavala, "An inertia-free filter line-search algorithm for large-scale nonlinear programming," *Comput. Optim. Appl.*, vol. 64, pp. 327–354, 2016.

[14] R. T. Rockafellar, "The multiplier method of hestenes and powell applied to convex programming," *J. Optim. Theory Appl.*, vol. 12, no. 6, pp. 555–562, 1973.

[15] A. R. Conn, N. I. Gould, and P. Toint, "A globally convergent augmented lagrangian algorithm for optimization with general constraints and simple bounds," *SIAM J. Numer. Anal.*, vol. 28, no. 2, pp. 545–572, 1991.

[16] F. E. Curtis, H. Jiang, and D. P. Robinson, "An adaptive augmented lagrangian method for large-scale constrained optimization," *Math. Program.*, vol. 152, nos. 1/2, pp. 201–245, 2015.

[17] P. E. Gill and D. P. Robinson, "A primal-dual augmented lagrangian," *Comput. Optim. Appl.*, vol. 51, no. 1, pp. 1–25, 2012.

[18] V. M. Zavala and M. Anitescu, "Real-time nonlinear optimization as a generalized equation," *SIAM J. Control Optim.*, vol. 48, no. 8, pp. 5444–5467, 2010.

[19] R. De Leone and O. L. Mangasarian, "Serial and parallel solution of large scale linear programs by augmented lagrangian successive overrelaxation," in *Optimization, Parallel Processing and Applications*. New York, NY, USA: Springer, 1988, pp. 103–124.

[20] A. R. Conn, G. Gould, and P. L. Toint, *LANCELOT: A Fortran Package for Large-Scale Nonlinear Optimization (Release A)*, vol. 17. New York, NY, USA: Springer, 2013.

[21] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, pp. 25–57, 2006.

[22] N.-Y. Chiang and V. M. Zavala, "An inertia-free filter line-search algorithm for large-scale nonlinear programming," *Comput. Optim. Appl.*, vol. 64, no. 2, pp. 327–354, 2014.

[23] R. Fletcher, N. Gould, S. Leyffer, P. Toint, and A. Wächter, "Global convergence of a trust-region SQP-filter algorithm for general nonlinear programming," *SIAM J. Optim.*, vol. 13, no. 3, pp. 635–659, 2002.

[24] S. J. Wright, "Superlinear convergence of a stabilized SQP method to a degenerate solution," *Comput. Optim. Appl.*, vol. 11, pp. 253–275, 1998.

[25] I. S. Duff and J. K. Reid, *MA27-a Set of Fortran Subroutines for Solving Sparse Symmetric Sets of Linear Equations*. Oxfordshire, U.K.: UKAEA Atomic Energy Res. Establishment, 1982.

[26] E. Anderson *et al.*, *LAPACK Users' Guide*, vol. 9. Philadelphia, PA, USA: SIAM, 1999.

[27] T. A. Davis, *User Guide for Cholmod: A Sparse Cholesky Factorization and Modification Package*, Dept. Comput. Inf. Sci. Eng., Univ. Florida, Gainesville, FL, USA, 2008.

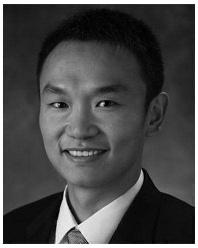
[28] I. Bongartz, A. R. Conn, N. I. M. Gould, and P. L. Toint, "CUTE: Constrained and unconstrained testing enviroment," *ACM Trans. Math. Soft.*, vol. 21, pp. 123–160, 1995.

[29] C. Greif, E. Moulding, and D. Orban, "Bounds on eigenvalues of matrices arising from interior-point methods," *SIAM J. Optim.*, vol. 24, no. 1, pp. 49–83, 2014.

Nai-Yuan Chiang received the B.E. degree in mathematics from Fudan University, Shanghai, China, in 2007, and the M.Sc. and Ph.D. degrees in optimization from the University of Edinburgh, Edinburgh, U.K., in 2008 and 2013, respectively.

He was a Postdoctoral Appointee at Argonne National Laboratory from 2013 to 2015. He is currently a Senior Research Engineer at the United Technologies Research Center, East Hartford, CT, USA. His current research interests include nonlinear optimization, high-performance computing and power system.

Dr. Chiang is also a member of Society for Industrial and Applied Mathematics, Institute for Operations Research and the Management Sciences, and Mathematical Optimization Society.



Rui Huang received the B.S. and M.S. degrees in control science and engineering from Zhejiang University, Hangzhou, China, in 2004 and 2006, respectively and the Ph.D degree in chemical engineering from Carnegie Mellon University, Pittsburgh, PA, USA, in 2010.

He is currently a Research Scientist at United Technologies Research Center. His research interests include advanced control, optimization applications, and optimal control formulations and algorithms.

Victor M. Zavala (M'09) received the B.Sc. degree from Universidad Iberoamericana, Mexico City, Mexico, and the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, USA, in 2008, both in chemical engineering.

He is the Richard H. Soit Assistant Professor in the Department of Chemical and Biological Engineering, University of Wisonsin-Madison, Madison, WI, USA. Before joining UW-Madison, he was a Computational Mathematician in the Mathematics and Computer Science Division, Argonne National Laboratory. . His research interests are in the areas of mathematical modeling of energy systems, high-performance computing, stochastic optimization, and predictive control.

Dr. Zavala is an Associate Editor for the *Journal of Process Control* and a Technical Editor of *Mathematical Programming Computation*.