2016 IEEE Conference on Control Applications (CCA)
Part of 2016 IEEE Multi-Conference on Systems and Control
September 19-22, 2016. Buenos Aires, Argentina

An Efficient SQP Algorithm for Moving Horizon Estimation with Huber Penalties and Multi-Rate Measurements

Dimitris Kouzoupis[1], Rien Quirynen[1,2], Fabian Girrbach[1,3] and Moritz Diehl[1]

***Abstract*— Moving Horizon Estimation (MHE) is a powerful, yet computationally expensive approach for state and parameter estimation that is based on online optimization. In applications with multi-rate measurements that may include outliers, the Huber penalty is often a better candidate for the MHE objective than the commonly used Euclidean norm. Treating this non-smooth objective in Newton-type optimization typically requires the use of slack variables that would in turn increase the problem size significantly. As an alternative, we propose a novel algorithm that combines Sequential Convex Programming (SCP) and Sequential Quadratic Programming (SQP) techniques in an effort to reduce the computational complexity. The proposed implementation is tailored to embedded applications, as it combines state-of-the-art numerical tools and efficient C code. We demonstrate the performance of the algorithm on a real-world state estimation problem where the position and orientation of a single propeller aircraft are estimated using GPS and IMU measurement data.**

I. INTRODUCTION

Moving Horizon Estimation (MHE) is a popular state and parameter estimation approach due to its intrinsic ability to handle constraints and nonlinear dynamics [1]. It requires the online solution of an Optimal Control Problem (OCP) that can be a major challenge for real-time implementations. However, recent algorithmic advancements in the field of optimal control have made the use of nonlinear MHE possible for systems with very fast dynamics [2], [3].

In many applications, measurements are provided from various sensors with different sampling rates and thus sensor fusion is necessary [4]. An efficient approach to implement real-time MHE with high-rate measurements was proposed by [5], where direct multiple shooting [6] was used in combination with collocation based integration schemes to treat the arising OCPs. The presented implementation relies on the auto generated integrators of the ACADO Toolkit [7] that use the continuous output feature to evaluate output functions at an arbitrary grid of the integration interval. The MHE problem formulation in [5] uses the Euclidean distance of the simulated values from the measurements in the objective which is a common choice in practice since the result can be interpreted as a maximum likelihood estimator for measurements with additive Gaussian noise [8]. However, measurements with outliers are better treated with different penalties that yield more robust estimators, e.g., the one norm or the Huber penalty function [9]. Such non-smooth objective functions are often reformulated with the use of slack variables [2]. The number of those variables is related to the total number of measurements which can be very large and therefore jeopardize the performance of the solver.

In this paper, we present a novel algorithm to solve a certain class of Nonlinear Programs (NLPs) that arise in MHE with Huber penalties. The method relies on several algorithmic components including multiple shooting for the discretization of the continuous time OCP, collocation based integrators with continuous output, Sequential Convex Programming (SCP) and finally Sequential Quadratic Programming (SQP) for the convex subproblems. Given an upper bound on the curvature of the convex function, we form a fixed Hessian approximation for the QP subproblems and prove that the resulting scheme is equivalent to a preconditioned proximal gradient method. Special focus has been given on the suitability of the algorithm for embedded applications by designing a prototype implementation whose time-critical components are based on efficient numerical tools and plain C code. Preliminary results on a real-world state estimation problem show that the proposed scheme can outperform general purpose sparse NLP solvers by an order of magnitude.

Section II introduces the general problem formulation that we wish to address and presents the SCP method. Section III proposes an SQP algorithm for the subproblems and proves its convergence properties. Section IV applies the derived algorithm to MHE with Huber penalties and multi-rate measurements. Section V discusses the software implementation while Section VI compares the performance of the proposed scheme to other approaches on a real-world state estimation example. Section VII concludes the paper.

II. PROBLEM STATEMENT

In this section we introduce the problem formulation of interest and discuss how SCP can be used to find locally optimal solutions.

A. The Nonlinear Program

We are interested in solving NLPs of the form:

$$\min_{w} \ \Psi(w) \tag{1a}$$

$$\text{s.t. } h(w) \leq 0, \tag{1b}$$

where $\Psi : \mathbb{R}^{n_f} \to \mathbb{R}$ is a convex function, piecewise twice continuously differentiable and $h : \mathbb{R}^{n_w} \to \mathbb{R}^{n_h}$ twice continuously differentiable but not necessarily convex. Note

[1]Department of Microsystems Engineering (IMTEK), University of Freiburg, Georges-Koehler-Allee 102, 79110 Freiburg, Germany.
[2]Department ESAT-STADIUS, KU Leuven University, Kasteelpark Arenberg 10, 3001 Leuven, Belgium.
[3]Xsens Technologies B.V., P.O. Box 559, 7500 AN Enschede, the Netherlands, www.xsens.com.

that equality constraints can be also expressed by (1b) with a suitable definition of $h(\cdot)$.

Assumption 1: The curvature of the convex function $\Psi(\cdot)$ is bounded from above, i.e.,

$$\hat{H} \succeq \nabla_w^2 \Psi(w) \tag{2}$$

for all w in the feasible set $\Omega := \{w \mid h(w) \leq 0\}$.

One approach for solving (1) is to use a general purpose NLP solver. Alternatively, one can apply SCP in order to exploit the existing convexity in the problem. This is the subject of the next section.

B. Sequential Convex Programming

An SCP method solves a series of convex subproblems in order to find a locally optimal solution of (1). It has locally linear convergence, as proven in [10]. In order to apply SCP on (1), we need to linearize at each iteration i the nonlinear function $h(\cdot)$ around the current iterate $w^{[i]} := \bar{w}$. This yields the following convex subproblem:

$$\min_{w} \ \Psi(w) \tag{3a}$$

$$\text{s.t. } Bw + b \leq 0, \tag{3b}$$

with $B = \frac{\partial h}{\partial w}(\bar{w})$ the Jacobian of the constraints and $b = h(\bar{w}) - B\bar{w}$ the constant term.

III. SOLVING THE CONVEX SUBPROBLEMS

Our aim is to solve instances of (3) efficiently for any convex function with a known upper bound on its curvature. A generic approach that does not require further knowledge on the nature of $\Psi(\cdot)$ is to apply SQP and solve a series of QP subproblems. At each iteration with iterate $w^{[i,j]} := \bar{\bar{w}}$, a Newton-type SQP method calculates a step direction by solving the QP:

$$\min_{\Delta w} \ \frac{1}{2} \Delta w^\top H(\bar{\bar{w}})\, \Delta w + \Delta w^\top \nabla \Psi(\bar{\bar{w}}) \tag{4a}$$

$$\text{s.t. } B\Delta w + B\bar{\bar{w}} + b \leq 0, \tag{4b}$$

where $\Delta w = w - \bar{\bar{w}}$ and $H(\bar{\bar{w}}) \approx \nabla_w^2 \Psi(w)$ denotes an approximation of the Hessian of the Lagrangian of (3). We chose here to express (3) in absolute and (4) in relative terms for notational convenience. Note also that the inequality constraints (3b) are affine and therefore identical to (4b).

A. A Fixed-Hessian SQP Method

Newton-type methods approximate the exact Hessian in various ways. Since each subproblem (3) is convex with affine constraints and an upper bound on the curvature of $\Psi(\cdot)$ is available, we can use this upper bound $\hat{H}$ to form a *constant* Hessian approximation. The following theorem proves the convergence properties of this scheme.

Theorem 2: Assume (3) has a global minimum $w^\star$ that satisfies the Linear Independence Constraint Qualification (LICQ) as defined in [11]. A full-step SQP method with fixed Hessian approximation $H \succ \frac{1}{2}\hat{H}$ converges globally to $w^\star$.

Proof: An SQP algorithm with fixed Hessian approximation is equivalent to a preconditioned proximal gradient method with fixed step size, as proved in Lemma 3 below [12], [13]. Moreover, this Hessian approximation yields a precondition matrix that satisfies the assumptions for global convergence guarantees [12]. ■

In contrast to other SQP methods, Theorem 2 shows that the proposed scheme based on the curvature bound $\hat{H}$ does not require line search or any other globalization strategy [11] to solve (3) from an arbitrary initial point.

B. Connection to First Order Methods

Let us define the generalized norm operator as $||w||_L = \langle Lw, w\rangle^{1/2}$ with $\langle\cdot,\cdot\rangle$ the inner product of two vectors. The projection of a point $\hat{w}$ onto the feasible set $\mathcal{W}$ with the aforementioned norm operator is defined by the optimization problem:

$$\min_{w} \ \frac{1}{2} \|w - \hat{w}\|_L^2 \tag{5a}$$

$$\text{s.t. } w \in \mathcal{W}, \tag{5b}$$

which we call a generalized projection with precondition matrix L [13]. The generalized proximal gradient method, as presented in [13], involves two steps at each iteration. Namely, a preconditioned gradient step followed by a generalized projection on the feasible set with the same precondition matrix. The equivalence between this first order scheme and the proposed fixed-Hessian SQP method is established by the following lemma.

Lemma 3: A fixed Hessian SQP method with full steps to solve (3) is equivalent to a preconditioned (or generalized) proximal gradient method with constant step size.

Proof: Problem (4) with a fixed Hessian approximation H is equivalent to:

$$\min_{w} \ \frac{1}{2} ||w - w^+||_H^2 \tag{6a}$$

$$\text{s.t. } Bw + b \leq 0, \tag{6b}$$

with $\Delta w = w - \bar{\bar{w}}$, $w^+ := \bar{\bar{w}} - H^{-1}\nabla\Psi(w)$ and a difference in the constant term that does not affect the optimal solution. Therefore, the intermediate iterate w^+ can be seen as the preconditioned gradient step and (6) as the generalized projection of this iterate on the feasible set, as in (5). ■

Note that the equivalence of Lemma 3 implies that accelerated variants of the presented algorithm can also be implemented, which differ only by basic arithmetic operations [12].

Remark 4: In the field of first order methods, having an expensive projection operator is typically avoided since one iteration of the algorithm becomes almost as expensive as the original problem. However, in our context, we can show that the overall cost of the SQP method is mainly determined by the first QP solution, while all subsequent iterations introduce a negligible overhead. More details on the computational complexity can be found in Section V.

C. Local Contraction of the Method

We have presented in the previous section a sufficient condition for the proposed Hessian approximation to be globally contractive, namely $H \succ \frac{1}{2}\hat{H}$. If no prior knowledge on the accuracy of the maximum curvature is available, H should be chosen equal to $\hat{H}$. On the other hand, if there exists an estimate on how much we overestimate this curvature, one could try to closer approximate the exact Hessian at the solution by choosing $\frac{1}{2}\hat{H} \prec H \prec \hat{H}$. This intuition is confirmed experimentally in Section VI.

IV. APPLICATION TO MHE

The algorithm of the previous section finds direct application in Moving Horizon Estimation (MHE) with Huber penalties. We demonstrate how to take into account the special structure of the problem when building the QP subproblems and discuss an alternative Hessian approximation.

A. MHE Problem Formulation

The Huber penalty for a scalar α is given by the function:

$$\phi_\rho(\alpha) = \begin{cases} \frac{1}{2}\alpha^2 & |\alpha| \leq \rho \\ \rho\left(|\alpha| - \frac{1}{2}\rho\right) & |\alpha| > \rho, \end{cases} \tag{7}$$

where the parameter $\rho \in \mathbb{R}$ defines the width of the quadratic region. For the multidimensional case, with $\alpha, \rho \in \mathbb{R}^M$, we define equivalently the vector:

$$\Phi_\rho(\alpha) = [\phi_{\rho_0}(\alpha_0), \ldots, \phi_{\rho_{M-1}}(\alpha_{M-1})]^\top, \tag{8}$$

that will be used to penalize the deviations from the measurements in the objective function.

A multiple shooting formulation of the MHE problem with multi-rate measurements and Huber penalties reads as:

$$\min_{x,u} \sum_{k=0}^{N-1} \omega_k^\top \, \Phi_{\rho_k}(R_k(x_k, u_k)) \tag{9a}$$

$$\text{s.t. } x_{k+1} = F_k(x_k, u_k), \quad k = 0, \ldots, N-1. \tag{9b}$$

Here N is the horizon length, $x_k \in \mathbb{R}^{n_x}$ the differential states and $u_k \in \mathbb{R}^{n_u}$ the inputs. The functions $F_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ denote the numerical simulation of the continuous-time dynamics. The M measurements of each shooting interval enter the objective via the nonlinear residual functions $R_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^M$ and are weighted with the corresponding element of $\omega_k \in \mathbb{R}^M_+$. Nonlinear stage constraints are omitted in (9) to simplify notation, as their presence would not affect the described algorithm. The same holds for the arrival cost that is typically added to the objective (9a) [1].

Applying SCP and linearizing the nonlinear functions $R_k(\cdot)$ and $F_k(\cdot)$ around the current iterate $\bar{x}, \bar{u}$ yields the convex subproblem:

$$\min_{x,u} \sum_{k=0}^{N-1} \omega_k^\top \Phi_{\rho_k}\left(C_k x_k + D_k u_k + e_k\right) \tag{10a}$$

$$\text{s.t. } x_{k+1} = A_k\, x_k + B_k\, u_k + c_k, \;\; k = 0, \ldots, N-1, \tag{10b}$$

with $C_k = \frac{\partial R_k}{\partial x_k}(\bar{x}_k, \bar{u}_k)$, $D_k = \frac{\partial R_k}{\partial u_k}(\bar{x}_k, \bar{u}_k)$ the sensitivities of the residual function with respect to states and inputs, $e_k = R_k(\bar{x}_k, \bar{u}_k) - C_k\, \bar{x}_k - D_k\, \bar{u}_k$ the corresponding constant term and the quantities A_k, B_k, c_k defined similarly for the continuity constraint in (9b). For later reference, we define the stage variables $z_k^\top = [x_k^\top, u_k^\top]$ and the concatenated sensitivities $E_k = [C_k,\, D_k]$.

Problem (10) can be converted into a QP using a smooth-reformulation of the Huber function [2]. However, this reformulation would introduce a large number of slack variables and the dimension of this QP would grow fast with the number of measurements. As a remedy, we suggest to apply the proposed fixed-Hessian SQP method and therefore we need to derive explicit expressions for the Hessian and gradient of the QP subproblems. This is the topic of the following section.

Remark 5: The Huber penalty as defined in (7) satisfies the assumptions of bounded curvature and piecewise twice differentiability, as introduced in Section II. For a discussion on convergence guarantees of the resulting semi-smooth Newton type scheme see, e.g., [14].

B. The Fixed-Hessian SQP Method for MHE

The proposed Hessian approximation can be evaluated cheaply using the linearization of the residuals and their corresponding weights. Since each term in the summation over N in (10a) depends only on the stage variables z_k, the Hessian matrix has a block diagonal structure. This implies that these blocks, as well as the corresponding parts of the gradient vector, can be calculated in parallel.

Taking into account the unit maximum curvature of the scalar Huber penalty as defined in (7), each Hessian block is formed as:

$$\hat{H}_k = {E_k}^\top \mathrm{diag}(\omega_k) E_k, \tag{11}$$

where $\mathrm{diag}(\omega_k) \in \mathbb{R}^{M \times M}$ is a symmetric matrix with the weights ω_k on the diagonal and zero elsewhere. Recall the earlier definition $E_k := [C_k \;\; D_k]$ with C_k, D_k the partial derivatives of the residual function. For each Hessian block $\hat{H}_k$, the corresponding gradient vector g_k is calculated by the following sum of contributions:

$$g_k = \sum_{m=1}^{M} \omega_m \rho_m \, \mathrm{sat}\left[\frac{1}{\rho_m}\left(E_{m,:}\bar{\bar{z}} + e_m\right)\right] {E_{m,:}}^\top, \tag{12a}$$

where we dropped the stage index k on the right-hand side for notational convenience and defined the saturation function as:

$$\mathrm{sat}[\alpha] = \begin{cases} 1 & \text{if } \alpha > \;\; 1 \\ -1 & \text{if } \alpha < -1 \\ \alpha & \text{otherwise.} \end{cases} \tag{12b}$$

The subscript m refers to the corresponding element of a vector while the double subscript $(m,:)$ denotes the m^{th} row of the respective matrix.

C. Online Improvement of the Hessian Approximation

An alternative approach for the Hessian approximation is to consider in (11) only the contributions from the residuals that belong to the quadratic region in the first SQP iteration (by setting the weights of outliers to zero). If this estimate happens to be optimal for the convex subproblem, our algorithm converges in one step as we identify the exact Hessian of a QP "in disguise". In any other case the algorithm may or may not perform better depending on how accurate this initial estimation is. Special care needs to be taken however, as an estimation of many outliers may lead to a Hessian approximation that does not respect the condition $H \succ \frac{1}{2}\hat{H}$. The performance of this approach is investigated in Section VI.

Remark 6: Updating the Hessian approximation at each SQP iteration to remove the contributions of current outliers would result in an exact Hessian SQP scheme. However, we will show in the next section that keeping the Hessian approximation fixed is a major advantage from a computational point of view, since subsequent QPs can be solved significantly faster.

V. SOFTWARE IMPLEMENTATION

The purpose of this section is to discuss some algorithmic details that make the presented algorithm suitable for embedded applications. Our implementation combines state-of-the-art optimization tools and efficient C code with deterministic runtime.

A. Integrators with Continuous Output

In order to evaluate and compute the derivatives of the nonlinear functions in (9b) we need tools for numerical integration and sensitivity propagation [7]. Given that the length of the shooting intervals in the NLP typically corresponds to the rate of the slow measurements, collocation based integrators that can evaluate additional outputs on a different grid are suitable to handle the available high-rate measurements. This motivates our choice to use the code generated ACADO integrators presented in [5] and freely available as part of the ACADO Toolkit [7].

B. Condensing

The equality constraints in (10b) uniquely determine the state trajectory as a function of the inputs and initial state. Therefore, one solution approach is to eliminate all state variables except for the first one and solve a smaller, dense QP instead. This procedure is also known as condensing and in combination with a dense QP solver it can outperform sparse or structure-exploiting solvers for certain problem dimensions. The most computationally expensive part in forming the condensed problem is the computation of the Hessian matrix which can be performed efficiently with the algorithm presented in [15].

Algorithm 1 The embedded SCP-SQP scheme

Input: $(x^{[0]}, u^{[0]})$, $i_{\max}, j_{\max}$, σ_{tol}, ϵ_{tol}
Output: Approximate solution $(x^{[i]}, u^{[i]})$ of (9)
for $i = 1 : i_{\max}$ **do**
  $x^{[i]} \leftarrow x^{[i-1]}$, $u^{[i]} \leftarrow u^{[i-1]}$
  for $k = 0 : N-1$ **do in parallel**
    Use integrator to get C_k, D_k, e_k and A_k, B_k, c_k
    Calculate Hessian block $\hat{H}_k$ (11)
  end for
  Calculate condensed Hessian
  for $j = 1 : j_{\max}$ **do**
    Calculate gradient (12)
    Calculate condensed gradient
    Solve dense QP and expand solution to get $(\Delta x, \Delta u)$
    Update $x^{[i]} \leftarrow x^{[i]} + \Delta x$, $u^{[i]} \leftarrow u^{[i]} + \Delta u$
    if $\|[\Delta x^\top, \Delta u^\top]\|_\infty \leq \epsilon_{\text{tol}}$ **then**
      $j = j_{\max}$
    end if
  end for
  if $\|[(x^{[i]} - x^{[i-1]})^\top, (u^{[i]} - u^{[i-1]})^\top]\|_\infty \leq \sigma_{\text{tol}}$ **then**
    Return $(x^{[i]}, u^{[i]})$
  end if
end for

C. Embedded QP solution

To solve the condensed QP efficiently we use the embedded QP solver qpOASES [16]. This open-source software tool is based on an online active-set strategy and it has been successfully used in many real-world applications [17]. The main motivation for using qpOASES comes from the fact that we can efficiently initialize the solver in consecutive QP subproblems. For that purpose, two initialization options are available. The first option is called *warm-starting* and it initializes the next QP with the solution of the previous problem. For active set solvers this guess can reduce the number of iterations significantly. The second option is called *hot-starting* and it additionally preserves all matrix factorizations. In the context of the presented scheme, the first QP of each SCP iteration is warm-started and all subsequent QPs are hot-started, leading to significantly faster execution times. This effect is also observed experimentally in Section VI.

D. The SCP-SQP Scheme for MHE

The complete algorithm is summarized in Algorithm 1. In each outer SCP iteration, numerical integrators are used (possibly in parallel) to linearize residual functions and dynamics. When the sparse QP is formed, the condensing algorithm calculates the condensed Hessian, which remains constant for the convex subproblem. In each inner SQP iteration, the exact gradient of the dense problem is formed, the QP is solved and the solution is expanded to the full space of state and input deviations. Both inner and outer loops stop when the step on the primal variables is below a certain tolerance or the maximum number of iterations is reached. Note here that for simplicity of notation and implementation, the inner iterates $z^{[i,j]} := z^{[i,j-1]} + \Delta z$ overwrite the current outer iterate $z^{[i]}$.

VI. EXPERIMENTAL RESULTS

To assess the performance of the presented MHE scheme, we consider the pose estimation of a single propeller aircraft during a take-off manoeuvre using both GPS and IMU measurements. The measurement data were acquired by the XSENS sensor MTi-G-700 at the rates of 4 Hz and 400 Hz respectively.

A. Model Equations

To describe the motion of the system, we use a rigid body model comprising 13 states and 6 inputs, namely:

$$x(t) = \begin{bmatrix} p^L(t) \\ \dot{p}^L(t) \\ \omega(t)^S_{LS} \\ q(t)^{LS} \end{bmatrix}, \quad u(t) = \begin{bmatrix} F^S(t) \\ \tau^S(t) \end{bmatrix}, \tag{13}$$

where $p^L \in \mathbb{R}^3$ and $\dot{p}^L \in \mathbb{R}^3$ denote the position and velocity with respect to the local frame, i.e., a frame fixed at the initial position of the body with unit vectors pointing east, north, up. The angular velocity from the sensor frame S to the local frame L is expressed in the sensor frame S and it is denoted by $\omega^S_{LS} \in \mathbb{R}^3$. It affects the orientation from frame S to L, expressed by the unit quaternion $q^{LS} \in \mathbb{R}^4$. The inputs to the system are the force $F^S \in \mathbb{R}^3$ and torque $\tau^S \in \mathbb{R}^3$, both expressed in the sensor frame S. The Ordinary Differential Equations (ODEs) that describe the trajectory of the rigid body read as:

$$\ddot{p}^L(t) = \frac{R^{LS}(t)F^S(t)}{m} \tag{14a}$$

$$\dot{\omega}^S_{LS}(t) = \left(I^S\right)^{-1}\left(\tau^S(t) - \omega^S_{LS}(t) \times \left(I^S\omega^S_{LS}(t)\right)\right) \tag{14b}$$

$$\dot{q}^{LS}(t) = \frac{1}{2}q^{LS}(t) \odot \omega^S_{LS}(t), \tag{14c}$$

with $R^{LS} \in \mathbb{R}^{3\times3}$ denoting the rotation matrix from S to L, $I^S \in \mathbb{R}^{3\times3}$ the constant inertia tensor of the rigid body in the S frame and m the mass of the rigid body. The operators $\times$ in (14b) and $\odot$ in (14c) denote the cross product and quaternion product respectively.

For the experiments that follow, we consider a multiple shooting discretization of the MHE problem similar to (9), with sampling time $T_s = 0.25\,\text{s}$ and horizon length $N = 30$. Given the two frequencies of the sensors, each shooting interval comprises $n_{\text{GPS}} = 1$ GPS and $n_{\text{IMU}} = 100$ IMU measurements. To form the residuals that penalize the deviations from the measurements, the following two output functions are used, evaluated at the corresponding rate of each sensor:

$$y_{\text{GPS}}(x(t), u(t)) = \begin{bmatrix} p^L(t) \\ \dot{p}^L(t) \end{bmatrix} \tag{15a}$$

$$y_{\text{IMU}}(x(t), u(t)) = \begin{bmatrix} R^{SL}(t)\left(\ddot{p}^L(t) - g^L\right) \\ \omega^S_{LS}(t) \end{bmatrix}. \tag{15b}$$

In (15b), g^L denotes the gravity vector with respect to L. An additional equality constraint is required in (9) to impose the unit norm of the quaternion at the beginning of the horizon. The dynamics of the system will then ensure that this constraint is satisfied for the whole trajectory at the optimal solution.

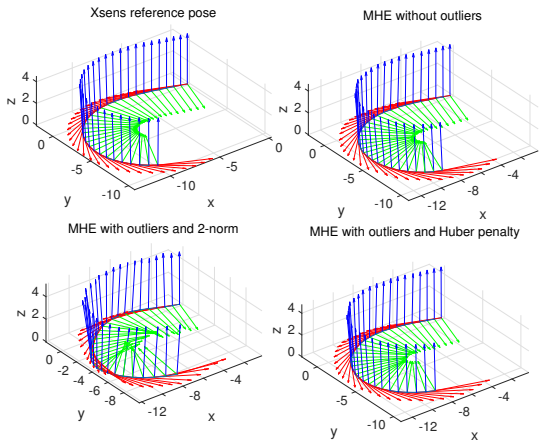

Fig. 1. Effect of outliers on the pose estimation.

B. Numerical Experiments

Let us first motivate the use of Huber penalties in MHE by means of an experiment. Figure 1 shows four different pose estimations for the same set of measurements. On the upper left part is the pose of the body as estimated by the on-board system of the sensor while on the right we plot the optimal trajectory as calculated by solving one MHE problem instance. The same set of measurements is then polluted with $n_{\text{O}} = 500$ artificial outliers and two different MHE problem formulations are compared. Namely, a formulation with quadratic penalties on the lower left and one with Huber penalties on the right. It is clear from the results that the Huber objective leads to a more robust pose estimation.

The main goal of the experiments is to illustrate the performance of our SCP-SQP scheme in Algorithm 1 and to compare it with alternative algorithmic approaches. As a first approach, we apply direct collocation and use the sparse NLP solver Ipopt [18], within casadi [19]. The Huber penalties are formulated as semi-smooth casadi functions avoiding the need of slack variables. The second approach is to apply SCP on the multiple shooting formulation (9) and solve the subproblems by converting (10) into a QP with the use of slack variables. Since each scalar Huber penalty function introduces two slacks as well as polyhedral inequality constraints and bounds, the size of these QPs grows rapidly with the number of measurements. MATLAB's native QP solver, quadprog, is used here to solve the subproblems. The remaining two approaches use the presented SCP-SQP scheme with fixed Hessian approximation. More precisely, we first use the same QP solver, quadprog, for the QP subproblems and then replace it with qpOASES. A comparison of all methods is summarized in Table I for different rates of IMU measurements (via downsampling). All simulations were performed on a 2,5 GHz Intel i7 processor with 16 GB of RAM, running OS X 10.11 and MATLAB R2015b.

The timings of Table I indicate that even with a general purpose tool like quadprog, we can improve the performance of the MHE estimator significantly by employing

TABLE I

TIME [S] UNTIL CONVERGENCE FOR DIFFERENT SCHEMES.

Method	f_{IMU} 100 Hz	200 Hz	400 Hz
casadi with Ipopt	0.84	1.35	3.07
SCP with quadprog	31.9	41.5	89.9
SCP-SQP with quadprog	0.43	0.29	0.31
SCP-SQP with qpOASES	0.11	0.10	0.14

TABLE II

DETAILED TIMINGS [MS] OF FIRST SCP ITERATION.

Operation	SQP + quadprog	SQP + qpOASES
Integration	19.61	19.54
Full condensing	-	1.38
First QP solution	10.58	1.18
Gradient condensing	-	0.07
Subsequent 4 QPs	42.83	0.69

the proposed SCP-SQP scheme. The performance of the latter remains almost unaffected by the number of IMU measurements per interval.

The improved performance of the tailored scheme is due to the fact that the total cost of the SQP iterations is mainly affected by the cost of the first QP solution. This is depicted in Table II where detailed timings of the first SCP iteration are reported for the two SCP-SQP schemes. The table is split in two parts. The first part comprises operations that are executed only once per convex subproblem while the operations of the second part are called as many times as the required number of SQP iterations. For the latter, the sum of all timings is reported. The condensing timings refer to a plain C implementation of the algorithm presented in [15].

To conclude the experiments, we compare the performance of the SCP-SQP method for the two proposed Hessian approximations of Section IV. Namely, the maximum curvature and the exact Hessian guess, as introduced in Section IV-C. Figure 2 shows that the second variant performs poorly in the first MHE iteration where the initialization of the solution is inaccurate while for the remaining optimization problems it converges consistently faster.

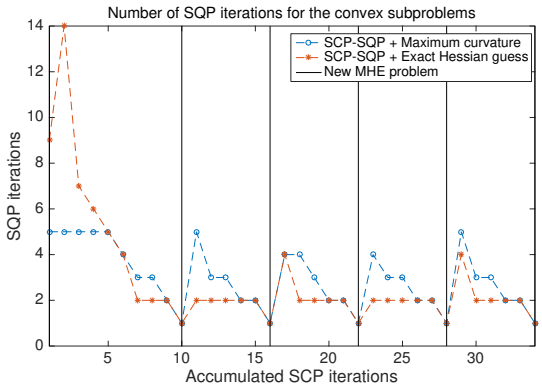

Fig. 2. SCP-SQP iterations for the two Hessian approximations.

VII. CONCLUSIONS

We have proposed an efficient algorithmic scheme to solve MHE problems with Huber penalties and multi-rate measurements. Instead of using a general purpose NLP solver on the original problem or solving high dimensional QPs within an SCP framework, our approach uses a nested SCP-SQP algorithm that solves the low-level QPs very efficiently. The performance of the scheme is demonstrated on a real-world state estimation problem.

ACKNOWLEDGMENTS

This research was supported by EU: FP7-TEMPO (MC ITN-607957), H2020-ITN AWESCO (642682), ERC HIGHWIND (259 166). R. Quirynen holds a PhD fellowship of the Research Foundation – Flanders (FWO).

REFERENCES

[1] C. V. Rao and J. B. Rawlings, "Nonlinear moving horizon state estimation," in *Nonlinear Predictive Control*, (Basel Boston Berlin), pp. 45–69, Birkhäuser, 2000.

[2] K. Geebelen, A. Wagner, S. Gros, J. Swevers, and M. Diehl, "Moving horizon estimation with a Huber penalty function for robust pose estimation of tethered airplanes," in *Proceedings of the American Control Conference (ACC)*, 2013.

[3] M. Vukov, S. Gros, G. Horn, G. Frison, K. Geebelen, J. B. Jørgensen, J. Swevers, and M. Diehl, "Real-time nonlinear MPC and MHE for a large-scale mechatronic application," *Control Engineering Practice*, 2015.

[4] R. López-Negrete and L. T. Biegler, "A moving horizon estimator for processes with multi-rate measurements: A nonlinear programming sensitivity approach," *Journal of Process Control*, vol. 22, no. 4, pp. 677–688, 2012.

[5] R. Quirynen, S. Gros, and M. Diehl, "Fast auto generated ACADO integrators and application to MHE with multi-rate measurements," in *Proceedings of the European Control Conference (ECC)*, pp. 3077–3082, 2013.

[6] H. G. Bock and K. J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," in *Proceedings of the IFAC World Congress*, pp. 242–247, Pergamon Press, 1984.

[7] R. Quirynen, M. Vukov, M. Zanon, and M. Diehl, "Autogenerating microsecond solvers for nonlinear MPC: a tutorial using ACADO integrators," *Optimal Control Applications and Methods*, vol. 36, pp. 685–704, 2014.

[8] A. Charnes, E. L. Frome, and P. L. Yu, "The equivalence of generalized least squares and maximum likelihood estimates in the exponential family," *Journal of the American Statistical Association*, vol. 71, no. 353, pp. 169–171, 1976.

[9] P. J. Huber, *Robust Statistics*. Wiley, 1981.

[10] Q. Tran-Dinh and M. Diehl, "Local convergence of sequential convex programming for nonconvex optimization," in *Recent advances in optimization and its application in engineering*, pp. 93–103, Springer-Verlag, 2010.

[11] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering, Springer, 2 ed., 2006.

[12] Y. Nesterov, *Introductory lectures on convex optimization: a basic course*, vol. 87 of *Applied Optimization*. Kluwer Academic Publishers, 2004.

[13] W. Zuo and Z. Lin, "A generalized accelerated proximal gradient approach for total-variation-based image restoration," *IEEE Transactions on Image Processing*, vol. 20, pp. 2748–2759, October 2011.

[14] L. Qi and J. Sun, "A nonsmooth version of Newton's method," *Mathematical Programming*, vol. 58, pp. 353–367, 1993.

[15] G. Frison and J. B. Jørgensen, "A fast condensing method for solution of linear-quadratic control problems," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, 2013.

[16] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl, "qpOASES: a parametric active-set algorithm for quadratic programming," *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[17] T. J. Besselmann, S. V. de moortel, S. Almér, P. Jörg, and H. J. Ferreau, "Model predictive control in the multi-megawatt range," *IEEE Transactions on Industrial Electronics*, 2015.

[18] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[19] J. Andersson, *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, K.U. Leuven, October 2013.