A New Continuous-Time Equality-Constrained Optimization to Avoid Singularity

Quan Quan and Kai-Yuan Cai

***Abstract*— In equality-constrained optimization, a standard regularity assumption is often associated with feasible point methods, namely, that the gradients of constraints are linearly independent. In practice, the regularity assumption may be violated. In order to avoid such a singularity, a new projection matrix is proposed based on which a feasible point method to continuous-time, equality-constrained optimization is developed. First, the equality constraint is transformed into a continuous-time dynamical system with solutions that always satisfy the equality constraint. Second, a new projection matrix without singularity is proposed to realize the transformation. An update (or say a controller) is subsequently designed to decrease the objective function along the solutions of the transformed continuous-time dynamical system. The invariance principle is then applied to analyze the behavior of the solution. Furthermore, the proposed method is modified to address cases in which solutions do not satisfy the equality constraint. Finally, the proposed optimization approach is applied to three examples to demonstrate its effectiveness.**

***Index Terms*— Continuous-time dynamical systems, equality constraints, optimization, singularity.**

I. Introduction

According to the implementation of a differential equation, most approaches to continuous-time optimization can be classified as either a dynamical system [1]–[3] or a neural network [4]–[10]. The dynamical system approach relies on the numerical integration of differential equations on a digital computer. Unlike discrete optimization methods, the step sizes of dynamical system approaches can be automatically controlled in the integration process and can sometimes be made larger than usual. This advantage suggests that the dynamical system approach can, in fact, be comparable with the currently available conventional discrete optimal methods and can facilitate faster convergence [1], [3]. The application of a higher order numerical integration process also enables us to avoid the zigzagging phenomenon, which is often encountered in typical linear extrapolation methods [1]. On the other hand, the neural network approach emphasizes implementation by analog circuits, very large-scale integration, and optical technologies [11]. The major breakthrough of this approach is attributed to [12], which introduced an artificial neural network to solve the traveling salesman problem. By employing analog hardware, the neural network approach offers low computational complexity and is suitable for parallel implementation.

For continuous-time equality-constrained optimization, existing methods can be classified into three categories [1], [4], [8], [13]: 1) the feasible point method (or primal method); 2) the penalty function method; and 3) the Lagrangian multiplier method (or primal-dual method). The feasible point method directly solves the original problem by searching through the feasible region for the optimal solution. Each point in the process is feasible, and the value of the objective function constantly decreases. Compared with the two other methods, the feasible point method offers three significant advantages that highlight its usefulness as a general procedure that is applicable to almost all nonlinear programming problems [13, p. 360]: 1) the terminating point is feasible if the process is terminated before the solution is reached; 2) the limit point of the convergent sequence of solutions must be at least a local constrained minimum; and 3) the method is applicable to general nonlinear programming problems because it does not rely on special problem structures, such as convexity. In equality-constrained optimization, a standard regularity assumption is often associated with feasible point methods, namely, that the gradients of constraints are linearly independent. Besides, the regularity assumption is also required by the penalty function method [13, p. 417] and the Lagrangian multiplier method [13, p. 476]. However, in practice, the regularity assumption may be violated.

In order to avoid such a singularity, a continuous-time feasible point method is proposed to identify the local minimum from a feedback control perspective for a general equality-constrained optimization problem. Compared with global optimization methods, local optimization methods are still necessary. First, they often serve as a basic component for some global optimizations, such as the branch-and-bound method [14]. On the other hand, they require less computation for online optimization. Compared with the discrete optimal methods offered by MATLAB, illustrative examples show that the proposed method avoids convergence to a singular point and facilitates faster convergence through numerical integration on a digital computer. Moreover, one illustrative example shows that the proposed projection matrix also outperforms the modified commonly used projection matrix. In view of these, the contributions of this paper are clear and listed as follows.

1) A new projection matrix is proposed to remove a standard regularity assumption that is often associated with feasible point methods, namely, that

Manuscript received July 15, 2014; accepted August 29, 2015. Date of publication September 22, 2015; date of current version January 18, 2016. This work was supported by the National Natural Science Foundation of China under Grant 61473012.

The authors are with the Department of Automatic Control, Beijing University of Aeronautics and Astronautics, Beijing 100191, China (e-mail: qq_buaa@buaa.edu.cn; kycai@buaa.edu.cn).

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/TNNLS.2015.2476348

the gradients of constraints are linearly independent (see [1, p. 158, eq. (4)], [2, p. 156, eq. (2.3)], [8, p. 1669, Assumption 1]). Compared with a modified commonly used projection matrix, the proposed projection matrix achieves a better precision. Moreover, its recursive form can be implemented more easily.

2) Based on the proposed projection matrix, a continuous-time, equality-constrained optimization method is developed to avoid convergence to a singular point. The invariance principle is applied to analyze the behavior of the solution.
3) The modified version of the proposed optimization is further developed to address cases in which solutions do not satisfy the equality constraint. This ensures its robustness against uncertainties caused by numerical error or realization by analog hardware.
4) Equality-constrained optimization is formulated as a control problem, in which the equality constraint is transformed into a continuous-time dynamical system. The proposed method is easily accessible, especially to practitioners in the control field.

The following notation is used. $\mathbb{R}^n$ is Euclidean space of dimension n. $\|\cdot\|$ denotes the Euclidean vector norm or induced matrix norm. I_n is the identity matrix with dimension n. $0_{n_1\times n_2}$ denotes a zero vector or a zero matrix with dimension $n_1\times n_2$. $(\cdot)^\dagger$ denotes the Moore–Penrose inverse. Direct product $\otimes$ and vec$(\cdot)$ operation are defined in Appendix A. The function $[\cdot]_\times : \mathbb{R}^3 \to \mathbb{R}^{3\times 3}$ with matrix $H \in \mathbb{R}^{9\times 3}$ is also defined in Appendix A. Suppose $g : \mathbb{R}^n \to \mathbb{R}$. The gradient of the function g is given by $\nabla g(x) = \nabla_x g(x) = [\partial g(x)/\partial x_1 \cdots \partial g(x)/\partial x_n]^T \in \mathbb{R}^n$ and the matrix of second partial derivatives of $g(x)$ known as Hessian is given by $\nabla_{xx} : \mathbb{R} \to \mathbb{R}^{n\times n}$ and $\nabla_{xx} g(x) = [\partial^2 g(x)/\partial x_i \partial x_j]_{ij}$.

II. Problem Formulation

In this section, the considered equality-constrained optimization problem is formulated first. Then, an equality constraint transformation is proposed to replace with the equality constraints. Based on it, the objectives are proposed.

A. Equality-Constrained Optimization

The class of equality-constrained optimization problems considered here is defined as follows:

$$\min_{x\in\mathbb{R}^n} v(x), \quad \text{s.t. } c(x) = 0_{m\times 1} \tag{1}$$

where $v : \mathbb{R}^n \to \mathbb{R}$ is the objective function and $c = [c_1\ c_2\ \cdots\ c_m]^T \in \mathbb{R}^m$, $c_i : \mathbb{R}^n \to \mathbb{R}$ are the equality constraints. They are both twice continuously differentiable. Denote by $\nabla c \triangleq [\nabla c_1, \nabla c_2, \ldots, \nabla c_m] \in \mathbb{R}^{n\times m}$. To avoid a trivial case, suppose the constraint (or feasible set) $\mathcal{F} = \{x \in \mathbb{R}^n | \, c(x) = 0_{m\times 1}\} \neq \emptyset$.

Definition 1 [15, pp. 316–317]: For the problem (1), a vector $x^* \in \mathcal{F}$ is a global minimum if $v(x^*) \le v(x)$, $\forall x \in \mathcal{F}$; a vector $x^* \in \mathcal{F}$ is a local (strict local) minimum if there is a neighborhood $\mathcal{N}$ of x^*, such that $v(x^*) \le v(x)$ $(v(x^*) < v(x))$ for $x \in \mathcal{N} \cap \mathcal{F}$.

Definition 2 [13, p. 325]: A vector $x^* \in \mathcal{F}$ is said to be a regular point if the gradient vectors $\nabla c_1(x^*), \nabla c_2(x^*), \ldots, \nabla c_m(x^*)$ are linearly independent. Otherwise, it is called a singular point.

Definition 3 [16, p. 117]: A function $v(x)$ satisfying $v(0) = 0$ and $v(x) > 0$ for $x \neq 0_{n\times 1}$ is said to be positive definite.

Remark 1 (On Inequality-Constrained Optimization): Inequality-constrained optimization problems can be transformed into equality-constrained optimization problems. For example, the inequality constraint $x \le 1, x \in \mathbb{R}$ can be replaced with an equality constraint $x + z^2 = 1, z \in \mathbb{R}$. On the other hand, the inequality constraint $-1 \le x \le 1, x \in \mathbb{R}$ can be replaced with an equality constraint $x = \sin(z), z \in \mathbb{R}$.

B. Equality Constraint Transformation

Optimization problems are often solved using numerical iterative methods. For an equality-constrained optimization problem, the major difficulty lies in ensuring that each iteration satisfies the constraints and can further move toward the minimum. To address this difficulty, a transformation of the equality constraints is proposed, which is first formulated as an assumption.

Assumption 1: For a given $x_0 \in \mathcal{F}$, there exists a function $f : \mathbb{R}^n \to \mathbb{R}^{n\times l}$, such that

$$\dot{x}(t) = f(x(t))u(t), \quad x(0) = x_0 \tag{2}$$

with solutions that satisfy $x(t) \in \mathcal{F}_u(x_0) \subset \mathcal{F}$, where $\mathcal{F}_u(x_0) = \{x(t) | \dot{x}(t) = f(x(t))u(t),\ x(0) = x_0 \in \mathcal{F},\ u(t) \in \mathbb{R}^l,\ t \ge 0\}$.

The best choice of $f(x)$ is to make $\mathcal{F}_u(x_0) = \mathcal{F}$. This holds for linear constraints. For the sake of simplicity, the variable t will be omitted except when necessary.

Theorem 1: Suppose that $c(x) = Ax$ and $f(x) \equiv A^\perp$, where $A^\perp$ is with full column rank, and the space spanned by the columns of $A^\perp$ is the null space of A. Then, $\mathcal{F}_u(x_0) = \mathcal{F}$, $\forall x_0 \in \mathcal{F}$.

Proof: See Appendix B. □

From the proof of Theorem 1, the choice of $f(x)$ is, in fact, an accessibility problem in the control field. However, it is difficult to achieve $\mathcal{F}_u(x_0) = \mathcal{F}$ in a general case. For example, if $c(x) = (x_1 + 1)(x_1 - 1)$, $x = [x_1\ x_2]^T \in \mathbb{R}^2$, then $\mathcal{F} = \{x \in \mathbb{R}^2 \big| \, x_1 = 1 \text{ or } x_1 = -1\}$. Since the two sets $\{x \in \mathbb{R}^2 \big| \, x_1 = 1\}$ and $\{x \in \mathbb{R}^2 \big| \, x_1 = -1\}$ are not connected, the solution of (2) starting from either set cannot access the other. Although $\mathcal{F}_u(x_0) \neq \mathcal{F}$, it is expected that the function $f(x)$ is chosen to make the set $\mathcal{F}_u(x_0)$ as large as possible, so that the probability of $x^* \in \mathcal{F}_u(x_0)$ is higher. Motivated by the linear case in Theorem 1, the function $f(x)$ should satisfy

$$\mathcal{V}_1(x) = \mathcal{V}_2(x)$$

where

$$\mathcal{V}_1(x) = \{z \in \mathbb{R}^n | \nabla c(x)^T z = 0_{m\times 1}\}, \quad [\text{null space of } \nabla c(x)^T]$$

$$\mathcal{V}_2(x) = \{z \in \mathbb{R}^n | z = f(x)u, u \in \mathbb{R}^l\}, \quad [\text{range space of } f(x)].$$

For a given $x' \in \mathbb{R}^n$, if $\mathcal{V}_1(x') = \mathcal{V}_2(x')$, then $f(x')$ is defined as the projection matrix of $\nabla c(x')$. Obviously, it satisfies

$$\nabla c(x')^T f(x') = 0_{m\times l}.$$

One commonly used projection matrix, denoted by $f_{\text{com}}(x)$, is given as follows [1], [2], [8]:

$$f_{\text{com}}(x) = I_n - (\nabla c(\nabla c^T \nabla c)^{-1} \nabla c^T)(x). \tag{3}$$

Obviously, the matrix function $f_{\text{com}}(x)$ is the projection matrix for all regular points except for some singular points.

C. Objective

This paper aims to propose a continuous-time, equality-constrained optimization method to identify the local minimum from a feedback control perspective. Concretely, the first objective is to propose a new projection matrix to avoid singularity. Based on it, the second objective is to design the update u to make the solutions of (2) achieve a local minimum. By Assumption 1, the update u in (2) can be considered as a control input from the feedback control perspective. The objective function $v(x)$ can be considered as a Lyapunov-like function, although $v(x)$ is, in fact, not required to be positive definite here. According to this, the second objective of this paper can be restated as: to design a control input u to decrease $v(x)$ along the solutions of (2) until x has achieved a local minimum.

III. New Projection Matrix

In order to avoid such a singularity, a new projection matrix is proposed. Before it, a modification of $f_{\text{com}}(x)$ in (3) is investigated.

A. Modified Commonly Used Projection Matrix

In order to avoid singularity problem, a commonly used projection matrix (3) is modified as follows:

$$f_{\text{mcom}}(x) = I_n - (\nabla c(\varepsilon I_m + \nabla c^T \nabla c)^{-1} \nabla c^T)(x) \tag{4}$$

where $\varepsilon > 0$ is a small positive scale. Obviously, the smaller ε is, the closer to zero $\|\nabla c^T f_{\text{mcom}}\|$ is. On the other hand, however, a very small ε will cause an ill-conditioning problem especially for a low-precision processor.

Example 1: Consider the following gradient vectors:

$$\begin{aligned}\nabla c_1 &= [1 \quad 1 \quad 1 \quad 1]^T\\ \nabla c_2 &= [2 \quad 1 \quad 1 \quad 1]^T\\ \nabla c_3 &= [3 \quad 2 \quad 2 \quad 2]^T\end{aligned} \tag{5}$$

which are linearly dependent as $\nabla c_3 = \nabla c_1 + \nabla c_2$. The precision error $e_p = \|\nabla c^T f_{\text{mcom}}\|$ is performed with different $\varepsilon = 10^{-k}, k = 1, \ldots, 15$, where $\nabla c = [\nabla c_1, \nabla c_2, \nabla c_3]$. As shown in Fig. 1, the smallest precision error is achieved only at $\varepsilon = 10^{-8}$ with a precision error around 10^{-8}. Reducing ε further will increase the numerical error.

In order to avoid the singularity problem in Example 1, the best cure is to remove the vector ∇c_3, resulting in

$$\nabla c_{\text{new}} = [\nabla c_1, \nabla c_2] \in \mathbb{R}^{4\times 2}.$$

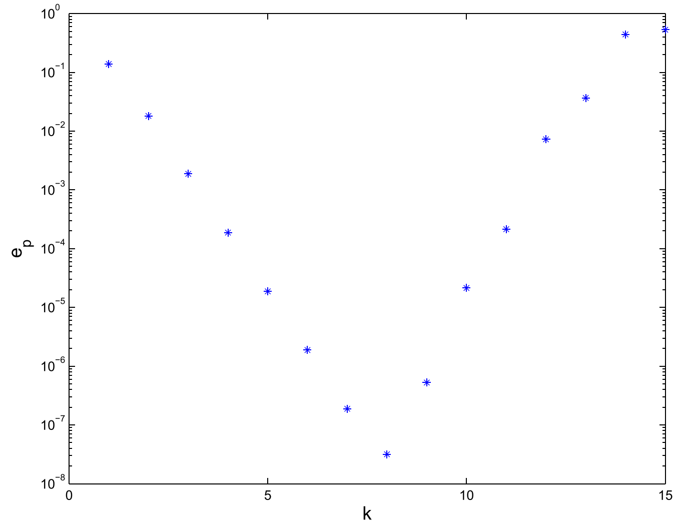


Fig. 1. Precision error of the modified commonly used projection matrix with different $\varepsilon = 10^{-k}$.

With it, the projection matrix becomes

$$f_{\text{newcom}} = I_4 - \nabla c_{\text{new}}\left(\nabla c_{\text{new}}^T \nabla c_{\text{new}}\right)^{-1} \nabla c_{\text{new}}^T.$$

It is easy to see that $\nabla c^T f_{\text{newcom}} \equiv 0_{3\times 4}$. However, the best cure cannot be implemented continuously using (3), which further cannot be realized by analog hardware. For such a purpose, a new projection matrix is proposed in the following.

B. New Projection Matrix

A new projection matrix, denoted by $f_{\text{pro}}(x)$, is proposed to avoid the singularity problem. For a special case $c : \mathbb{R}^n \to \mathbb{R}$, such $f_{\text{pro}}(x)$ is designed in Theorem 2. Consequently, a method is proposed to construct a projection matrix for a general case $c : \mathbb{R}^n \to \mathbb{R}^m$. Before the design, the following preliminary result is needed.

Lemma 1: Suppose

$$\begin{aligned}\mathcal{W}_1 &= \{z \in \mathbb{R}^n | L^T z = 0\}\\ \mathcal{W}_2 &= \left\{z \in \mathbb{R}^n | z = \left(I_n - \frac{LL^T}{\delta(\|L\|^2) + \|L\|^2}\right) u, u \in \mathbb{R}^n\right\}\end{aligned}$$

where $L \in \mathbb{R}^n$ and

$$\delta(x) = \begin{cases} 1, & x = 0, x \in \mathbb{R}\\ 0, & x \neq 0, x \in \mathbb{R}.\end{cases}$$

Then, $\mathcal{W}_1 = \mathcal{W}_2$.

Proof: See Appendix C. □

Theorem 2: Suppose that $c : \mathbb{R}^n \to \mathbb{R}$ and the function $f_{\text{pro}}(x)$ is designed to be

$$f_{\text{pro}} = I_n - \frac{\nabla c \nabla c^T}{\delta(\|\nabla c\|^2) + \|\nabla c\|^2}. \tag{6}$$

Then, Assumption 1 is satisfied with $u \in \mathbb{R}^n$ and $\mathcal{V}_1(x) = \mathcal{V}_2(x)$, $\forall x \in \mathbb{R}^n$.

Proof: Since $\dot{c}(x) = \nabla c(x)^T \dot{x}$ and $\dot{x} = f_{\text{pro}}(x)u$, it follows $\dot{c}(x) \equiv 0_{m\times 1}$ by Lemma 1. Therefore, Assumption 1 is satisfied with $u \in \mathbb{R}^n$. Further by Lemma 1, $\mathcal{V}_1(x) = \mathcal{V}_2(x)$. □

Theorem 3: Suppose that $c : \mathbb{R}^n \to \mathbb{R}^m$ and the functions $f_k(x)$ are in a recursive form as follows:

$$f_0 = I_n$$
$$f_k = f_{k-1} - \frac{f_{k-1} f_{k-1}^T \nabla c_k \nabla c_k^T f_{k-1}}{\delta\left(\| f_{k-1}^T \nabla c_k\|^2\right) + \| f_{k-1}^T \nabla c_k\|^2} \tag{7}$$

where $k = 1, \ldots, m$. Then, Assumption 1 is satisfied with $f_{\text{pro}} = f_m$ and $u \in \mathbb{R}^n$ and $\mathcal{V}_1(x) = \mathcal{V}_2(x)$, $\forall x \in \mathbb{R}^n$.

Proof: See Appendix D. □

Remark 2 (On the Best Cure): If ∇c_m is represented by a linear combination of ∇c_i, then $\nabla c \nabla c^T$ is singular, $i = 1, \ldots, m-1$. In this case, $\nabla c_m^T f_{m-1} = 0_{1\times n}$ as $\nabla c_i^T f_{m-1} = 0_{1\times n}$, $i = 1, \ldots, m-1$. By (7), the projection matrix f_{pro} will degenerate to $f_{\text{pro}} = f_{m-1}$, that is equivalent to removing the term ∇c_m. This is consistent with the best cure aforementioned.

Remark 3 (On Example 1): Let us revisit Example 1. In practice, the impulse function $\delta(x)$ is approximated by some continuous functions, such as $\delta(x) \approx e^{-\gamma|x|}$, where γ is a large positive scale. The precision error $e_p = \|\nabla c^T f_{\text{pro}}\|$ is performed with $\delta(x) \approx e^{-30|x|}$, resulting in $e_p = 2.7629 \times 10^{-10}$. This demonstrates the advantage of our proposed projection matrix over f_{mcom} in (4). Furthermore, compared with (3) or (4), the explicit recursive form of the proposed projection matrix is also easier to realize by analog hardware because of avoiding the matrix inversion.

IV. Update Design and Convergence Analysis

In this section, the update (or say controller) u is designed to make $\dot{v}(x) \le 0$, so that $v(x)$ is nonincreasing. If $v(x)$ is positive definite, then the theories of Lyapunov stability is available. That will be very familiar to practitioners in the control field. However, in order to make $v(x)$ more general, the objective function $v(x)$ here is not required to be positive definite or convexity. Because of this, the analysis is based on the LaSalle invariance theorem [16, pp. 126–129].

A. Update Design

Based on Assumption 1, taking the time derivative of $v(x)$ along the solutions of (2) results in

$$\dot{v}(x) = \nabla v(x)^T f(x) u \tag{8}$$

where $\nabla v(x) \in \mathbb{R}^n$. In order to get $\dot{v}(x) \le 0$, a direct way of designing u is proposed as follows:

$$u = -Q(x) f(x)^T \nabla v(x) \tag{9}$$

where $Q : \mathbb{R}^n \to \mathbb{R}^{l\times l}$ and $Q(x) \ge \epsilon I_l > 0$, $\epsilon > 0$, $\forall x \in \mathbb{R}^n$. Then, (8) becomes

$$\dot{v}(x) = -\nabla v(x)^T f(x) Q(x) f(x)^T \nabla v(x) \le 0. \tag{10}$$

Substituting (9) into the continuous-time dynamical system (2) results in

$$\dot{x} = -f(x) Q(x) f(x)^T \nabla v(x) \tag{11}$$

with solutions which always satisfy the constraint $c(x) = 0_{m\times 1}$. The closed-loop system corresponding to the continuous-time dynamical system (2) and the controller (9) is shown in Fig. 2.

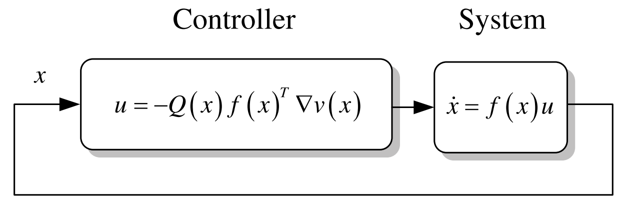


Fig. 2. Closed-loop control system.

B. Convergence Analysis

The invariance principle is applied to analyze the behavior of the solution of (11). The readers not interested in these details can directly proceed to Section IV-C.

Theorem 4: Under Assumption 1, given $x_0 \in \mathcal{F}$, if the set $\mathcal{K} = \{x \in \mathbb{R}^n | v(x) \le v(x_0), c(x) = 0_{m\times 1}\}$ is bounded, then the solution of (11) starting at x_0 approaches $x_l^* \in \mathcal{S}$, where $\mathcal{S} = \{x \in \mathcal{K} | \nabla v(x)^T f(x) = 0_{1\times l}\}$. In addition, if $\mathcal{V}_1(x_l^*) = \mathcal{V}_2(x_l^*)$, then there must exist $\lambda^* = [\lambda_1^* \ \lambda_2^* \cdots \lambda_m^*]^T \in \mathbb{R}^m$, such that $\nabla v(x_l^*) = \sum_{i=1}^m \lambda_i^* \nabla c_i(x_l^*)$ and $c(x_l^*) = 0_{m\times 1}$, namely, x_l^* is a Karush–Kuhn–Tucker (KKT) point. Furthermore, if $z^T \nabla_{xx} L(x_l^*, \lambda^*) z > 0$, for all $z \in \mathcal{V}_1(x_l^*), z \ne 0_{n\times 1}$, then x_l^* is a strict local minimum, where $L(x, \lambda) = v(x) - \sum_{i=1}^m \lambda_i c_i(x)$.

Proof: The proof is composed of three propositions. Proposition 1 is to show that $\mathcal{K}$ is compact and positively invariant with respect to (11). Proposition 2 is to show that the solution of (11) starting at x_0 approaches $x_l^* \in \mathcal{S}$. Proposition 3 is to show that $x_l^* \in \mathcal{S}$ is a KKT point, further a strict local minimum. The three propositions are proven in Appendix E. □

Corollary 1: Suppose that $f(x) = f_{\text{pro}}(x)$ as in (7) for $c : \mathbb{R}^n \to \mathbb{R}^m$ and the set $\mathcal{K} = \{x \in \mathbb{R}^n | v(x) \le v(x_0), c(x) = 0_{m\times 1}\}$ is bounded for given $x_0 \in \mathcal{F}$. Then, the solution of (11) starting at x_0 approaches $x_l^* \in \mathcal{S}$, where $\mathcal{S} = \{x \in \mathcal{K} | \nabla v(x)^T f(x) = 0_{1\times n}\}$ and x_l^* is a KKT point. In addition, if $z^T \nabla_{xx} L(x_l^*, \lambda^*) z > 0$, for all $z \in \mathcal{V}_1(x_l^*), z \ne 0_{n\times 1}$, then x_l^* is a strict local minimum, where $L(x, \lambda) = v(x) - \sum_{i=1}^m \lambda_i c_i(x)$.

Proof: Since $\mathcal{V}_1(x_l^*) = \mathcal{V}_2(x_l^*)$ by Theorem 3, the remainder of the proof is the same as that of Theorem 4. □

Corollary 2: Consider the following equality-constrained optimization problem:

$$\min_{x\in\mathbb{R}^n} v(x), \quad \text{s.t. } Ax = b \tag{12}$$

where $v(x)$ is convex and twice continuously differentiable, $A \in \mathbb{R}^{m\times n}$ with rank $A < n$, and $\mathcal{K} = \{x \in \mathbb{R}^n | v(x) \le v(x_0), Ax = b\}$ is bounded, then the solution of (11) with $f(x) \equiv A^{\perp}$ starting at any $x_0 \in \mathcal{F}$ approaches the global minimum x^*.

Proof: The solution of (11) starting at x_0 approaches $x_l^* \in \mathcal{S}$. Since rank $A < n$, it holds that $\mathcal{V}_1(x_l^*) = \mathcal{V}_2(x_l^*) \ne \varnothing$. Since the equality-constrained optimization (12) is convex, a KKT point x_l^* is the global minimum x^* of the problem (12). The remainder of the proof is the same as that of Theorem 4. □

Remark 4 (On Boundedness of Set $\mathcal{K}$): If $\mathcal{K}$ is not a bounded set, then $\mathcal{S}$ defined in Theorem 4 may be empty.

Therefore, the boundedness of the set $\mathcal{K}$ is necessary. For example, $v(x) = x_1 + x_2$, show that $c(x) = x_1 - x_2 = 0$. The set $\mathcal{K} = \{x \in \mathbb{R}^2 \big| x_1 + x_2 \leq v(x_0), x_1 - x_2 = 0\}$ is unbounded. According to Theorem 1, $f(x) \equiv [1\ 1]^T$. In this case, $\nabla v(x)^T f(x) \equiv 2 \neq 0$, and then the set $\mathcal{S}$ is empty.

C. Modified Closed-Loop Dynamical System

Although the proposed method ensures that the solutions satisfy the constraints, this method may fail if $x_0 \notin \mathcal{F}$ or if numerical algorithms are used to compute the solutions. Moreover, if the impulse function δ in (6) is approximated, then the constraints will also be violated. With these facts, the following modified closed-loop dynamical system is proposed to amend this situation.

Similar to [2], a correction term $-\rho\nabla c(x)c(x)$ is introduced into (11), resulting in

$$\begin{aligned}\dot{x} &= -\rho\nabla c(x)c(x) - f(x)Q(x)f(x)^T\nabla v(x)\\ x(0) &= x_0\end{aligned} \tag{13}$$

where $\rho > 0$. Define $v_c(x) = c(x)^T c(x)$. Then

$$\dot{v}_c(x) = -\rho c(x)^T\nabla c(x)^T\nabla c(x)c(x) \leq 0$$

where $\nabla c(x)^T f(x) \equiv 0_{m\times n}$ is utilized. Therefore, the solutions of (13) will tend to the feasible set $\mathcal{F}$ if $\nabla c(x)$ is of full column rank. Once $c(x) = 0_{m\times 1}$, the modified dynamical system (13) degenerates to (11). The self-correcting feature enables the step size to be automatically controlled in the numerical integration process or to tolerate uncertainties when the differential equation is realized using analog hardware. Singular points will make $\nabla c(x)$ be of nonfull column rank. In this case, the self-correction will not occur in the entire space with respect to $c(x)$, namely, $\|Dc(x)\| \to 0$ not $\|c(x)\| \to 0$, where $D \in \mathbb{R}^{p\times m}$, $p < m$. After passing singular points, the self-correction works in the entire space again. Once $\|c(x)\| = 0$, the property is mainly maintained by the proposed projection matrix.

Remark 5 (On the Matrix Q(x)): The matrix $Q(x)$ plays a role in avoiding instability in the numerical solution of differential equations by normalizing the Lipschitz condition of functions, such as $f(x)Q(x)f(x)^T\nabla v(x)$. If the dimension of dynamical system (13) is low, then let $Q(x) = \mu/\|f(x)f(x)^T\nabla v(x)\|$ for simplicity, where $\mu > 0$. The matrix $Q(x)$ also plays a role in coordinating the convergence rate of all states by minimizing the condition number of the matrix functions, such as $f(x)Q(x)f(x)^T$, especially in the case that the dimension of $f(x)Q(x)f(x)^T$ is high. Ideally, it is expected $f(x)Q(x)f(x)^T = I_n$. However, it is difficult to obtain such $Q(x)$, since the number of degrees of freedom of $Q(x) \in \mathbb{R}^{l\times l}$ is less than the number of elements of I_n. A natural choice is proposed as follows:

$$Q(x) = \mu(f(x)^T f(x))^\dagger + \epsilon I_l$$

where $\mu, \epsilon > 0$. The term ϵI_l makes $Q(x) \geq \epsilon I_l > 0$. As a result, one has

$$f(x)Q(x)f(x)^T = \mu U\Sigma\Sigma^\dagger\Sigma^\dagger\Sigma U^T + \epsilon f(x)f(x)^T.$$

Here, the singular value decomposition of the matrix $f(x)$ is $f(x) = U\Sigma V$, where $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{l\times l}$ are real unitary matrices, and $\Sigma \in \mathbb{R}^{n\times l}$ is a rectangular diagonal matrix with non-negative real numbers on the diagonal. If the dynamical system (13) is solved by the numerical integration of differential equations, $Q(x)$ needs to be computed every time. This, however, will cost much time. A time-saving way is to update $Q(x)$ at a reasonable interval. On the other hand, if the dynamical system (13) is realized by a neural network, then $Q(x)$ should be realized by an additional continuous-time dynamical system. Thanks to parallel implementation, the realization can also be implemented online. Finally, the matrix function $Q(x)$ also plays a role in determining the search direction. In the future, it is expected that the matrix function $Q(x)$ can be designed to generate search directions like those the interior point method and the conjugate gradient method provide.

Remark 6 (On Large-Scale Optimization): Consider the following equality-constrained optimization problem:

$$\min_{x\in\mathbb{R}^n} v(x), \quad \text{s.t. } c(x) = c'(x) + c_0 = 0_{m\times 1}$$

where $c_0 \in \mathbb{R}^m$ is the constant vector depending on a concrete problem. The continuous-time dynamical system (13) can be written as

$$\dot{x} = -\rho\nabla c(x)(c'(x) + c_0) + h(x)$$

where $h(x) = -f(x)Q(x)f(x)^T\nabla v(x) \in \mathbb{R}^n$. For such a class of large-scale optimization problems, the terms $h(x)$ are the same for any given $c_0 \in \mathbb{R}^m$. In order to make the method efficient, the function $h(x)$ can be derived offline first and realized by analog hardware. As shown in Fig. 3(a), $h(\cdot)$ is a block by taking x as an input and $h(x)$ as an output. Therefore, $h(x)$ depends on x, but $h(\cdot)$ does not. Thanks to the recursive form, as shown in Fig. 3(b), the realization of the proposed projection matrix $f(\cdot)$ can use the element $F(\cdot,\cdot)$ repeatedly. This makes the realization easier. The realization of $F(\cdot,\cdot)$ is shown in Fig. 3(c). In practice, the term $h(x)$ can be approximated offline by a static and simple neural network $n_n(x)$ [simpler than $h(x)$ at least] if only a bound $x \in \mathcal{D}$ is considered, resulting in

$$\dot{x} = -\rho\nabla c(x)c(x) + n_n(x).$$

Although the approximation error exists, the solution can still converge to the constraints because of the correction term $-\rho\nabla c(x)c(x)$.

V. Illustrative Examples

Three examples are given. The first example is mainly to show, in the presence of singularity, a commonly-used method will fail to find the minimum, whereas the proposed method can. The second example is mainly to show that the proposed projection matrix outperforms the modified commonly used projection matrix. The third example is mainly to show how the proposed method is applied to a practical example. Two of the three examples also show the advantage in running time.

TABLE I
RESULT FOR EXAMPLE IN SECTION V-A

Method	Initial Point	Solution	Optimal Value	cpu time (sec.)
Matlab fmincon	[-3 1]^T	[-1 -1]^T	2.0000	Not Available
New method	[-3 1]^T	[0.2062 -0.8546]^T	0.7729	0.125
Matlab fmincon	[2 -4]^T	[-1 -1]^T	2.0000	Not Available
New method	[2 -4]^T	[0.2062 -0.8545]^T	0.7726	0.0940
Matlab fmincon	[1 -4]^T	[0.2143 -0.8533]^T	0.7740	0.2030
New method	[1 -4]^T	[0.2056 -0.8550]^T	0.7733	0.1100

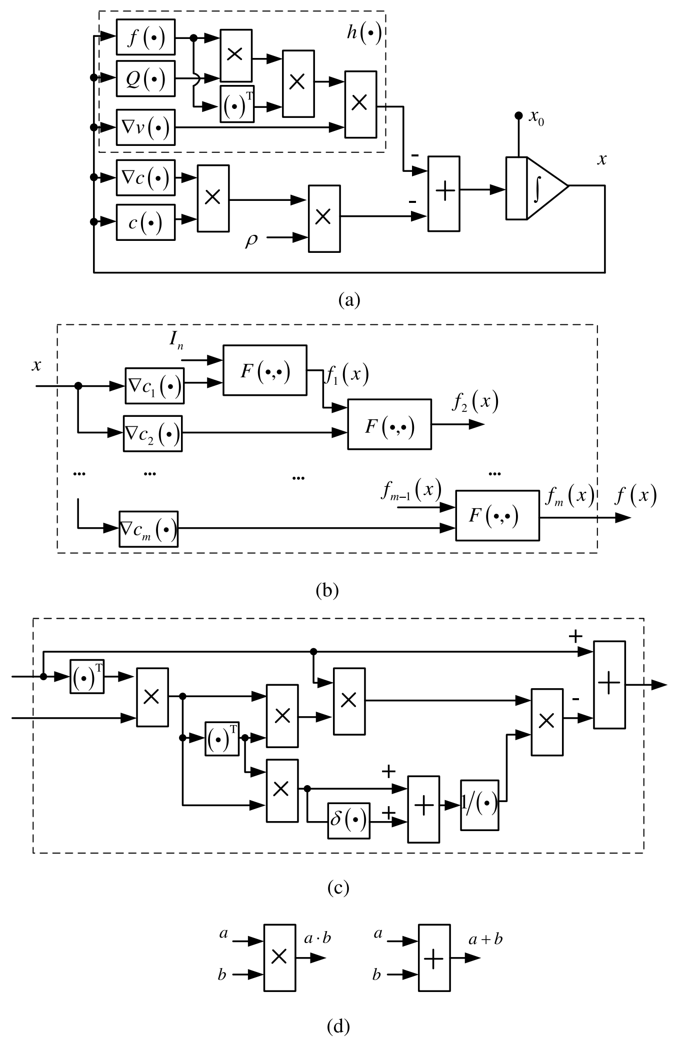


Fig. 3. Block diagram realization of (a) (13), (b) new projection matrix $f(\cdot)$, (c) element $F(\cdot,\cdot)$, and (d) multiplier and sum.

A. Estimate of Attraction Domain

For a given Lyapunov function, the crucial step in any procedure for estimating the attraction domain is determining the optimal estimate. Consider the system of differential equations

$$\dot{x} = Ax + g(x) \tag{14}$$

where $x \in \mathbb{R}^n$ is the state vector, $A \in \mathbb{R}^{n\times n}$ is a Hurwitz matrix, and $g : \mathbb{R}^n \to \mathbb{R}^n$ is a vector function. Let $v(x) = x^T Px$ be a given quadratic Lyapunov function for the origin of (14), i.e., $P \in \mathbb{R}^{n\times n}$ is a positive-definite matrix, such that $A^T P + PA < 0_{n\times n}$. Then, the largest ellipsoidal estimate of the attraction domain of the origin can be computed via the following equality-constrained optimization problem [17]:

$$\min_{x\in\mathbb{R}^n\backslash\{0_{2\times 1}\}} x^T Px, \quad \text{s.t. } x^T P[Ax + g(x)] = 0.$$

Since $\{x \in \mathbb{R}^n | x^T Px \le x_0^T Px_0\}$ is bounded, the subset

$$\mathcal{K} = \{x \in \mathbb{R}^n | x^T Px \le x_0^T Px_0, x^T P[Ax + g(x)] = 0\}$$

is bounded no matter what g is. For simplicity, consider (14) with $x = [x_1\ x_2]^T \in \mathbb{R}^2$, $A = -I_2$, $P = I_2$ and $g(x) = (c(x) + 1)[x_1\ x_2]^T$, where $c(x) = (x_1 + x_2 + 2)((x_2 + 1) - 0.1(x_1 + 1)^2)$. Then, the optimization problem is formulated as

$$\min_{x\in\mathbb{R}^2\backslash\{0_{2\times 1}\}} x_1^2 + x_2^2, \quad \text{s.t. } \left(x_1^2 + x_2^2\right)c(x) = 0.$$

Since $x \ne 0_{2\times 1}$, the problem is further formulated as

$$\min_{x\in\mathbb{R}^2} v(x) = x_1^2 + x_2^2, \quad \text{s.t. } c(x) = 0.$$

Then

$$\begin{aligned}\nabla v(x) &= [2x_1\ 2x_2]^T \\ \nabla c(x) &= \begin{bmatrix} d_2 - 0.1d_1^2 - 0.2d_1d_3 \\ d_2 - 0.1d_1^2 + d_3 \end{bmatrix} \\ d_1 &= x_1 + 1, \quad d_2 = x_2 + 1, \quad d_3 = x_1 + x_2 + 2.\end{aligned}$$

In this example, the modified dynamical system (13) is adopted, where f is chosen as f_{pro} in (6) with $\delta(x) = e^{-\gamma|x|}$, and $\gamma = 10, \rho = Q = 20/\|\nabla cc - f_{\text{pro}} f_{\text{pro}}^T \nabla v\|$. It is then solved using the MATLAB function ode45 with variable step.[1] The comparisons with the MATLAB optimal constrained nonlinear multivariate function fmincon are derived in Table I. The point $x_s = [-1\ -1]^T$ is a singular point, at which $\nabla c(x_s) = [0\ 0]^T$. As shown in Table I, under initial points $[-3\ 1]^T \in \mathcal{F}$ and $[2\ -4]^T \in \mathcal{F}$, by the MATLAB function, the singular point instead of the minimum is found, whereas the proposed method still finds the minimum. Under initial point $[1\ -4]^T \notin \mathcal{F}$, the proposed method can still find the minimum, similar to the MATLAB function. Under a different initial value, the evolutions of (13) are shown in Fig. 4. As shown, once close to the singular point $[-1\ -1]^T$, the solutions

[1]In this section, all computation is performed by MATLAB 6.5 on a personal computer (Asus x8ai) with Intel core Duo 2 Processor at 2.2 GHz.

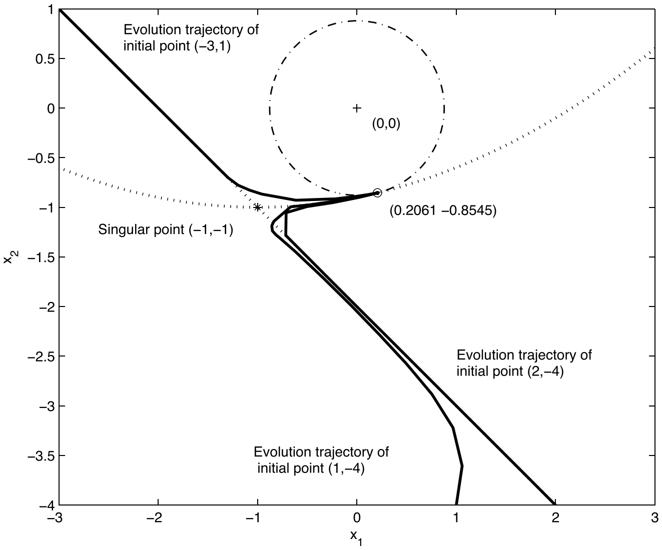
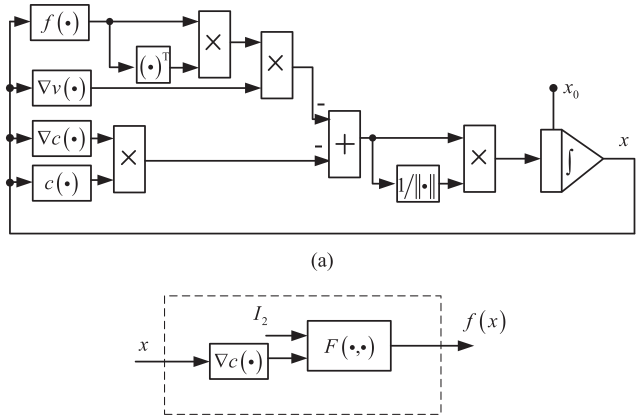


Fig. 4. Optimization of the estimate of attraction domain. Solid line: solution evolution. Dotted line: constraint evolution. Dashed–dotted line: objective evolution.



(a)

(b)

Fig. 5. Block diagram realization for example in Section V-A. (a) Block diagram realization. (b) Block diagram realization of the new projection matrix.

of (13) change direction and then move to the minimum $x_l^* = [0.2061 \quad -0.8545]^T$. Compared with the discrete optimal methods offered by MATLAB, these results show that the proposed method avoids convergence to a singular point. Moreover, the proposed method is comparable with currently available conventional discrete optimal methods and facilitates even faster convergence. The latter conclusion is consistent with that proposed in [1] and [3]. The realization by analog hardware is shown in Fig. 5.

B. Optimization With Minimum Being Singular Point

Consider a special equality-constrained optimization problem as follows:

$$\min_{x\in\mathbb{R}^3} v(x) = x_1^2 + x_2^2 + \left(x_3 - \frac{1}{\sqrt{3}}\right)^2$$
$$\text{s.t. } c_1(x) = x_1 + x_2 + x_3 - 1$$
$$c_2(x) = x_1 + x_2 + x_3^3 - 1 + \frac{1}{\sqrt{3}} - \frac{1}{(\sqrt{3})^3}.$$

The minimum is

$$x^* = \left[\frac{1}{2}\left(1-\frac{1}{\sqrt{3}}\right) \frac{1}{2}\left(1-\frac{1}{\sqrt{3}}\right) \ \frac{1}{\sqrt{3}}\right]^T.$$

Meanwhile, the gradients of the constraints are

$$\nabla c_1 = \begin{bmatrix}1\\1\\1\end{bmatrix}, \quad \nabla c_2(x) = \begin{bmatrix}1\\1\\3x_3^2\end{bmatrix}.$$

Obviously, $\nabla c_1(x^*) = \nabla c_2(x^*)$, namely, the minimum is also the singular point. Therefore, if the solution is closer to the minimum, the projection matrix $f_{\text{com}}(x)$ in (3) is more singular. Hence, the modified projection matrix f_{mcom} in (4) is considered. The proposed projection matrix is constructed in the following. By (7), f_1 is obtained as

$$f_1 = I_3 - \frac{\nabla c_1 \nabla c_1^T}{\delta(\|\nabla c_1\|^2) + \|\nabla c_1\|^2} = \begin{bmatrix} \frac{2}{3} & -\frac{1}{3} & -\frac{1}{3} \\ -\frac{1}{3} & \frac{2}{3} & -\frac{1}{3} \\ -\frac{1}{3} & -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

where $\delta(\|\nabla c_1\|^2) = 0$ as ∇c_1 is constant. Furthermore, the proposed projection matrix is written as

$$f_{\text{pro}}(x) = f_1 - \frac{f_1 f_1^T \nabla c_2(x) \nabla c_2^T(x) f_1}{\varepsilon + \|f_1^T \nabla c_2(x)\|^2} \tag{15}$$

where $\delta(\|f_1^T \nabla c_2(x)\|^2)$ is replaced by ε same to that in f_{mcom} for comparison. In particular, since $\nabla c_2^T(x^*) f_1 = 0_{1\times 3}$ at the singularity point, one has

$$f_{\text{pro}}(x^*) = f_1$$

which still possesses $\nabla c(x^*)^T f_{\text{pro}}(x^*) = 0_{2\times 3}$ no matter what ε is. In this example, the modified dynamical system (13) is adopted, where $x_0 = [-3 \ -2 \ (1/\sqrt{3})]^T$ and $\gamma = 10$, $\rho = Q = 20/\|\nabla cc - ff^T \nabla v\|$, $f = f_{\text{mcom}}, f_{\text{pro}}$. The differential equation (13) is solved using the MATLAB function ode45 with variable step. Denote $x_{\text{est}}(k)$ to be the stable solution when $\varepsilon = 10^{-k}$, $k = 1, \ldots 10$. The results are shown in Fig. 6, where $c(x) = [c_1(x) \ c_2(x)]^T$. As shown in Fig. 6 (first subplot), the obtained solutions using f_{pro} are closer to the minimum than those solutions using f_{mcom}. Moreover, as shown in Fig. 6 (second subplot), the obtained solutions using f_{pro} are further constrained better. These observation can be explained that, at the singularity point, $\nabla c(x^*)^T f_{\text{pro}}(x^*) = 0_{2\times 3}$, whereas $\nabla c(x^*)^T f_{\text{mcom}}(x^*) \neq 0_{2\times 3}$. Finally, it is observed that the advantage of the proposed projection matrix exists if $\varepsilon \geq 10^{-4}$. This is also useful because a larger ε implies an easier realization by lower-precision hardware, vice versa.

C. Estimate of Rotation Matrix

1) Problem Formulation: For simplicity, assume that images are taken by two identical pinhole cameras with the focal length equal to one. As shown in Fig. 7, the two cameras are specified by the camera centers $C_1, C_2 \in \mathbb{R}^3$ and attached

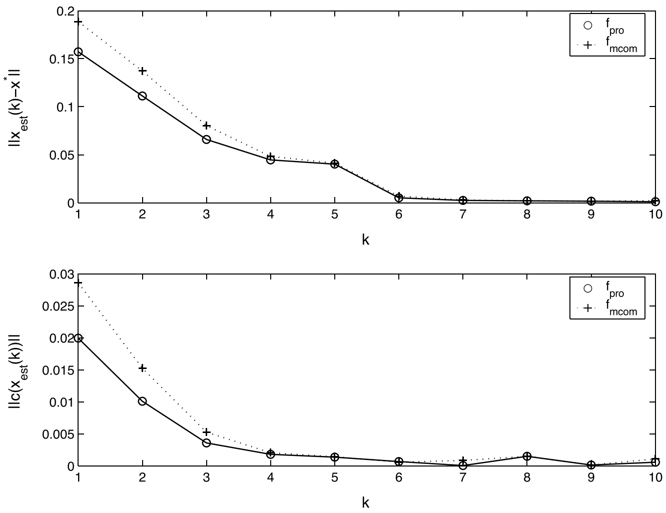


Fig. 6. Optimization by applying two projection matrices $f_{\text{mcom}}, f_{\text{pro}}$ to (13), respectively.

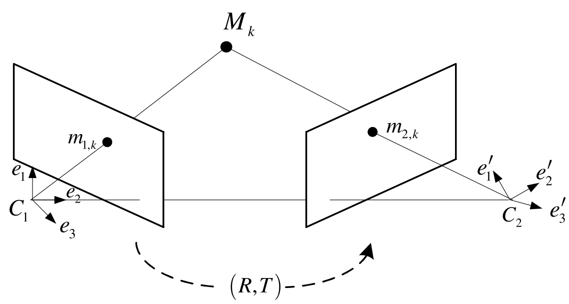


Fig. 7. Epipolar geometry.

orthogonal camera frames $\{e_1, e_2, e_3\}$ and $\{e_1', e_2', e_3'\}$, respectively. Denote $T = C_2 - C_1 \in \mathbb{R}^3$ to be the translation from the first camera to the second and $R \in \mathbb{R}^{3\times 3}$ to be the rotation matrix from the basis vectors $\{e_1, e_2, e_3\}$ to $\{e_1', e_2', e_3'\}$, expressed with respect to the basis $\{e_1, e_2, e_3\}$. Then, it is well known in the computer vision literature [18] that two corresponding image points are represented as follows:

$$m_{1,k} = \frac{1}{M_k(3)} M_k$$
$$m_{2,k} = \frac{1}{M_k'(3)} M_k', \quad k = 1, 2, \ldots, N \tag{16}$$

where M_k and M_k' represent the positions of the kth point expressed in the two camera frames $\{e_1, e_2, e_3\}$ to $\{e_1', e_2', e_3'\}$, respectively; $M_k(3)$ and $M_k'(3)$ represent the third element of vectors M_k, M_k', respectively. They have the relationship $M_k = RM_k' + T$, $k = 1, 2, \ldots, N$. These corresponding image points satisfy the epipolar constraints [18, p. 257]

$$m_{1,k}^T E m_{2,k} = 0, \quad k = 1, 2, \ldots, N \tag{17}$$

where $E = [T]_\times R$ is known as the essential matrix.

Using the direct product $\otimes$ and the vec$(\cdot)$ operation, the equations in (17) are equivalent to

$$A\varphi = 0_{N\times 1} \tag{18}$$

where

$$A = \begin{bmatrix} m_{2,1}^T & \otimes & m_{1,1}^T \\ & \vdots & \\ m_{2,N}^T & \otimes & m_{1,N}^T \end{bmatrix} \in \mathbb{R}^{N\times 9}$$
$$\varphi = \text{vec}([T]_\times R). \tag{19}$$

In practice, these image points $m_{1,k}$ and $m_{2,k}$ are subject to noise, $k = 1, 2, \ldots, N$. Therefore, T and R are often solved by the following optimization problem:

$$\min_{x\in\mathbb{R}^{12}} \upsilon(x) = \frac{1}{2}\varphi(x)^T A^T A \varphi(x)$$
$$\text{s.t. } \frac{1}{2}(\|T\|^2 - 1) = 0$$
$$\frac{1}{2}(R^T R - I_3) = 0_{3\times 3} \tag{20}$$

where $x = [T^T \ \text{vec}(R)^T]^T \in \mathbb{R}^{12}$. This is an equality-constrained optimization considered here. In the following, the proposed method is applied to the optimization problem (20).

2) Projection Matrix: By Theorem 2, the projection matrix for the constraint $(1/2)(\|T\|^2 - 1) = 0$ is

$$f_{\text{pro},1} = I_3 - \frac{TT^T}{\delta(\|T\|^2) + \|T\|^2}.$$

Since $\|T\|^2 = 1$ has to be satisfied exactly or approximately, then $\delta(\|T\|^2) = 0$. Therefore, the projection matrix for the constraint is

$$f_{\text{pro},1} = I_3 - TT^T/\|T\|^2.$$

Then, the constraint is transformed into

$$\dot{T} = (I_3 - TT^T/\|T\|^2)u_1$$

where $u_1 \in \mathbb{R}^3$. The constraint $(1/2)(R^T R - I_3) = 0_{3\times 3}$ is transformed into

$$\dot{R} = [u_2]_\times R \tag{21}$$

where $u_2 \in \mathbb{R}^3$. If so, then

$$\frac{d}{dt}(R^T R) = R^T \dot{R} + \dot{R}^T R$$
$$= R^T\left([u_2]_\times + [u_2]_\times^T\right)R = 0_{3\times 3}.$$

Furthermore, (21) is rewritten as

$$\text{vec}(\dot{R}) = (R^T \otimes I_3)Hu_2.$$

Then, the continuous-time dynamical system, whose solutions always satisfy the equality constraints $(1/2)(\|T\|^2 - 1) = 0$ and $(1/2)(R^T R - I_3) = 0_{3\times 3}$, is expressed as (2) with

$$f_{\text{pro}}(x) = \begin{bmatrix} f_{\text{pro},1} & 0_{3\times 3} \\ 0_{9\times 3} & (R^T \otimes I_3)H \end{bmatrix} \in \mathbb{R}^{12\times 6}$$
$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \in \mathbb{R}^6. \tag{22}$$

If the initial value $\|T(0)\|^2 = 1$ and $R(0)^T R(0) = I_3$, then all solutions of (2) satisfy the equality constraints.

3) Update Design: Since $\nabla v(x) = [(R^T \otimes I_3)H \; I_3 \otimes [T]_\times]^T A^T A\varphi$, the time derivative of $v(x)$ along the solutions of (2) is

$$\dot{v}(x) = -\varphi^T A^T A \Theta(x)^T Q(x)\Theta(x) A^T A\varphi \le 0$$

where

$$\Theta(x) = \begin{bmatrix} (I_3 - TT^T/\|T\|^2)^T H^T & (R^T \otimes I_3)^T \\ H^T (R^T \otimes I_3)^T & (I_3 \otimes [T]_\times)^T \end{bmatrix} \in \mathbb{R}^{6\times 9}.$$

The simplest way of choosing $Q(x)$ is $Q(x) \equiv I_6$. In this case, the eigenvalues of the matrix $A\Theta(x)^T\Theta(x)A^T$ are often ill-conditioned, namely

$$\lambda_{\min}(A\Theta(x)^T\Theta(x)A^T) \ll \lambda_{\max}(A\Theta(x)^T\Theta(x)A^T).$$

Convergence rates of the components of $A\varphi(x)$ depend on the eigenvalues of $A\Theta(x)^T Q(x)\Theta(x)A^T$. As a consequence, some components of $A\varphi$ converge fast, while the other may converge slowly. This leads to poor asymptotic performance of the closed-loop system. It is expected that each component of $A\varphi$ can converge at the same speed as far as possible. Suppose that there exists $\bar{Q}(x)$, such that

$$A\Theta(x)^T \bar{Q}(x)\Theta(x)A^T = I_9.$$

Then

$$\dot{v}(x) = -\varphi^T A^T A\varphi \le 0.$$

By Theorem 4, x will approach the set $\{x \in \mathbb{R}^n | \, A\varphi(x) = 0_{N\times 1}\}$, each element of which is a global minimum since $v(x) = 0$ in the set. Moreover, each component of $A\varphi$ converges at a similar speed. However, it is difficult to obtain such $\bar{Q}(x)$, since the number of degrees of freedom of $\bar{Q}(x) \in \mathbb{R}^{6\times 6}$ is less than the number of elements of I_9. A modified way is to make $A\Theta(x)^T Q(x)\Theta(x)A^T \approx I_9$. A natural choice is proposed as follows:

$$Q(x) = \mu(\Theta(x)A^T A\Theta(x)^T)^\dagger + \epsilon I_6 \qquad (23)$$

where $\mu > 0$, $(\Theta(x)A^T A\Theta(x)^T)^\dagger$ denotes the Moore–Penrose inverse of $\Theta(x)A^T A\Theta^T(x)$. The matrix ϵI_6 is to make $Q(x)$ positive definite, where ϵ is a small positive real. From the procedure above, $(\Theta(x)A^T A\Theta(x)^T)^\dagger$ needs to be computed every time. This, however, will cost much time. A time-saving way is to update $Q(x)$ at a reasonable interval. Then, (11) becomes

$$\dot{x} = -f_{\text{pro}}(x)Q(x)\Theta(x)A^T A\varphi(x) \qquad (24)$$

where $f_{\text{pro}}(x)$ is defined in (22), and $Q(x)$ is defined in (23). The differential equation can be solved by the Runge–Kutta methods. The solutions of (24) satisfy the constraints, where $x = [T^T \; \text{vec}(R)^T]^T$. Moreover, the dynamic system will reach some final resting state eventually.

4) Simulation: Suppose that there exist six points in the field of view, whose positions are expressed in the first camera frame as follows: 1) $M_1 = [-1 \; 1 \; 1]^T$; 2) $M_2 = [2 \; 0 \; 1]^T$; 3) $M_3 = [1 \; -1 \; 1]^T$; 4) $M_4 = [-1 \; -1 \; 1]^T$; 5) $M_5 = [1 \; 1 \; 1]^T$; and 6) $M_6 = [-1 \; 3 \; 1]^T$. Compared with the first camera frame, the second camera frame has translated and rotated with

$$\bar{T} = \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix}, \quad \bar{R} = \begin{bmatrix} 0.9900 & -0.0894 & 0.1088 \\ 0.0993 & 0.9910 & -0.0894 \\ -0.0998 & 0.0993 & 0.9900 \end{bmatrix}.$$

TABLE II

RESULT FOR EXAMPLE IN SECTION V-C

Method	$\|R^{*T}\bar{R} - I_3\|$	cpu time (sec.)
MATLAB fmincon	1.2469e-004	0.2500
New Approach	1.8784e-005	0.1400

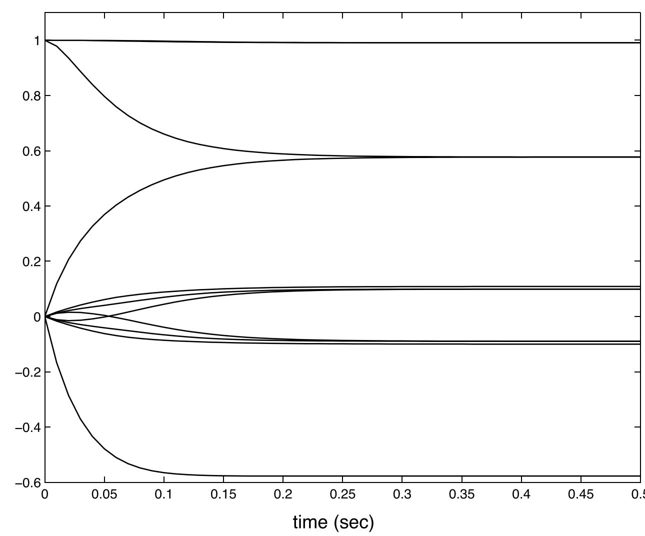


Fig. 8. Evolvement of the state.

The image points are generated by (16). Using the generated image points, A is obtained by (19). Set the initial value as follows: $T(0) = [0 \; 0 \; 1]^T$, $R(0) = I_3$, $\mu = 20$, and $\epsilon = 0.2$. The differential equation (24) is solved using MATLAB function ode45 with variable step. Compared with MATLAB optimal constrained nonlinear multivariate function fmincon, the following comparisons are given.

As shown in Table II, the proposed method requires less time to achieve a higher accuracy. Given that $v(x^*) = 0$, the solution is a global minimum. The evolution of each element of x is shown in Fig. 8. The state eventually reaches a rest state at a similar speed. With different initial values, several other simulations are also implemented. Based on the results, the proposed algorithm has met the expectations.

VI. CONCLUSION

A method to continuous-time, equality-constrained optimization based on a new projection matrix is proposed for the determination of local minima. With the transformation of the equality constraint into a continuous-time dynamical system, the class of equality-constrained optimization is formulated as a control problem. The resultant method is more general than the existing control theoretic approaches. Thus, the proposed method serves as a potential bridge between the optimization and control theories. Compared with other standard discrete-time methods, the proposed method avoids convergence to a singular point and facilitates faster convergence through numerical integration on a digital computer.

Appendix

A. Kronecker Product Vector and Skew-Symmetric Matrix

The symbol $\text{vec}(X)$ is the column vector obtained by stacking the second column of X under the first, and then the third, and so on. With $X = [x_{ij}] \in \mathbb{R}^{n\times m}$, the Kronecker product $X \otimes Y$ is the matrix

$$X \otimes Y = \begin{bmatrix} x_{11}Y & \cdots & x_{1m}Y \\ \vdots & \ddots & \vdots \\ x_{n1}Y & \cdots & x_{nm}Y \end{bmatrix}.$$

The relation $\text{vec}(XYZ) = (Z^T \otimes X)\text{vec}(Y)$ holds [19, p. 318]. The cross product of two vectors $x \in \mathbb{R}^3$ and $y \in \mathbb{R}^3$ is denoted by $x \times y = [x]_\times y$, where the symbol $[\cdot]_\times : \mathbb{R}^3 \to \mathbb{R}^{3\times3}$ is defined as [21, p. 194]

$$[x]_\times \triangleq \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \in \mathbb{R}^{3\times3}.$$

By the definition of $[x]_\times$, $x \times x = [x]_\times x = 0_{3\times1}$, $\forall x \in \mathbb{R}^3$ and

$$\begin{aligned} \text{vec}([x]_\times) &= Hx \\ H &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \end{aligned}$$

B. Proof of Theorem 1

Since $\mathcal{F}_u(x_0) \subseteq \mathcal{F}$, the remaining task is to prove $\mathcal{F} \subseteq \mathcal{F}_u(x_0)$, $\forall x_0 \in \mathcal{F}$, namely, for any $\bar{x} \in \mathcal{F}$, there exists a control input $u \in \mathbb{R}^l$ that can transfer any initial state $x_0 \in \mathcal{F}$ to $\bar{x}$. Since $x_0, \bar{x} \in \mathcal{F}$, there exist $u_0, \bar{u} \in \mathbb{R}^l$, such that $\bar{x} = A^\perp \bar{u}$ and $x(0) = A^\perp u_0$ by the definition of $A^\perp$. Design a control input

$$u(t) = \begin{cases} \frac{1}{\bar{t}}(\bar{u} - u_0), & 0 \le t \le \bar{t} \\ 0_{l\times1}, & t > \bar{t}. \end{cases}$$

With the control input above, it holds that

$$\begin{aligned} x(t) - x(0) &= \int_0^t A^\perp u(s)ds \\ &= \int_0^{\bar{t}} A^\perp u(s)ds = A^\perp \bar{u} - A^\perp u_0 \end{aligned}$$

when $t \ge \bar{t}$. Then, $x(t) = \bar{x}$, $t \ge \bar{t}$. Hence $\mathcal{F} \subseteq \mathcal{F}_u(x_0)$, $\forall x_0 \in \mathcal{F}$. Consequently, $\mathcal{F} = \mathcal{F}_u(x_0)$, $\forall x_0 \in \mathcal{F}$.

C. Proof of Lemma 1

Since

$$\begin{aligned} \delta(\|L\|^2) + \|L\|^2 &= 1, \quad \text{if } L = 0_{n\times1} \\ \delta(\|L\|^2) + \|L\|^2 &= \|L\|^2, \quad \text{if } L \ne 0_{n\times1} \end{aligned}$$

one has $\delta(\|L\|^2) + \|L\|^2 \ne 0$, $\forall L \in \mathbb{R}^n$. According to this, the following relationship holds:

$$\begin{aligned} &L^T(I_n - LL^T/(\delta(\|L\|^2) + \|L\|^2)) \\ &\quad = L^T - L^T\|L\|^2/(\delta(\|L\|^2) + \|L\|^2) \\ &\quad \equiv 0 \quad \forall L \in \mathbb{R}^n. \end{aligned}$$

This implies that $L^T z = 0$, $\forall z \in \mathcal{W}_2$, namely, $\mathcal{W}_2 \subseteq \mathcal{W}_1$. On the other hand, any $z \in \mathcal{W}_1$ is rewritten as

$$z = (I_n - LL^T/(\delta(\|L\|^2) + \|L\|^2))z$$

where $L^T z = 0$ is utilized. Hence $\mathcal{W}_1 \subseteq \mathcal{W}_2$. Consequently, $\mathcal{W}_1 = \mathcal{W}_2$.

D. Proof of Theorem 3

The proof is done by mathematical induction. Denote

$$\begin{aligned} \mathcal{V}_1^j &= \{z \in \mathbb{R}^n | \nabla c_i^T z = 0, i = 1, \ldots, j, j \le m\} \\ \mathcal{V}_2^j &= \{z \in \mathbb{R}^n | z = f_j u_j, u_j \in \mathbb{R}^n, j \le m\}. \end{aligned}$$

First, by Theorem 2, it is easy to see that the conclusions are satisfied with $j = 1$. Suppose that $\mathcal{V}_1^{k-1} = \mathcal{V}_2^{k-1}$ holds. Then, prove that $\mathcal{V}_1^k = \mathcal{V}_2^k$ holds. If so, this proof is concluded.

By $\mathcal{V}_1^{k-1}(x) = \mathcal{V}_2^{k-1}(x)$, one has

$$\begin{aligned} \mathcal{V}_1^k &= \{z \in \mathbb{R}^n | \nabla c_k^T z = 0, z \in \mathcal{V}_1^{k-1}\} \\ &= \{z \in \mathbb{R}^n | \nabla c_k^T z = 0, z = f_{k-1}u_{k-1}, u_{k-1} \in \mathbb{R}^n\} \\ &= \{z \in \mathbb{R}^n | \nabla c_k^T f_{k-1}u_{k-1} = 0, z = f_{k-1}u_{k-1}, u_{k-1} \in \mathbb{R}^n\}. \end{aligned}$$

On the other hand, by Lemma 1, one has

$$\begin{aligned} &\nabla c_k^T f_{k-1}u_{k-1} = 0 \\ &\quad \Leftrightarrow u_{k-1} = \left(I_n - \frac{f_{k-1}^T \nabla c_k \nabla c_k^T f_{k-1}}{\delta(\|f_{k-1}^T \nabla c_k\|^2) + \|f_{k-1}^T \nabla c_k\|^2}\right)u_k \end{aligned}$$

namely

$$\mathcal{V}_1^k = \mathcal{V}_2^k = \{z \in \mathbb{R}^n | z = f_k u_k, u_k \in \mathbb{R}^n\}$$

where

$$f_k = f_{k-1}\left(I_n - \frac{f_{k-1}^T \nabla c_k \nabla c_k^T f_{k-1}}{\delta(\|f_{k-1}^T \nabla c_k\|^2) + \|f_{k-1}^T \nabla c_k\|^2}\right).$$

E. Proof of Propositions in Theorem 4

The proof of Theorem 4 uses three propositions. Their proofs are shown as follows.

1) *Proof of Proposition 1:* In the space $\mathbb{R}^n$, the set $\mathcal{K}$ is compact if and only if it is bounded and closed in [20, p. 41, Th. 8.2]. Hence the remainder of this paper is to prove that $\mathcal{K}$ is closed. Suppose, to the contrary, $\mathcal{K}$ is not closed. Then, there exists a sequence $x(t_n) \in \mathcal{K} \to p \notin \mathcal{K}$ with $t_n \to \infty$. Whereas, $v(p) = \lim_{t_n\to\infty} v(x(t_n)) \le v(x_0)$ and $c(p) =$

$\lim_{t_n \to \infty} c(x(t_n)) = 0_{m\times 1}$ which imply $p \in \mathcal{K}$. The contradiction implies that $\mathcal{K}$ is closed. Hence, the set $\mathcal{K}$ is compact. By (10), $v(x) \le v(x_0)$ with respect to (11), $t \ge 0$. By Assumption 1, all solutions of (11) satisfy $c(x) = 0_{m\times 1}$. Therefore, $\mathcal{K}$ is positively invariant with respect to (11).

2) *Proof of Proposition 2:* Since $\mathcal{K}$ is compact and positively invariant with respect to (11), by Theorem 4.4 (invariance principle) in [16, p. 128], the solution of (11) starting at x_0 approaches $\dot{v}(x) = 0$, namely, $\nabla v(x)^T f(x) = 0$. In addition, since (11) becomes $\dot{x} = 0_{n\times 1}$ in $\mathcal{S}$, the solution approaches a constant vector $x_l^* \in \mathcal{S}$.

3) *Proof of Proposition 3:* Since $\mathcal{V}_1(x_l^*) = \mathcal{V}_2(x_l^*)$ and $x_l^* \in \mathcal{S}$ satisfy the following two equalities:

$$\nabla v(x_l^*)^T f(x_l^*) = 0_{1\times n}, \quad c(x_l^*) = 0_{m\times 1}$$

there exists u, such that $z = f(x_l^*)u$ for any $z \in \mathcal{V}_1(x_l^*)$. As a consequence, for any $z \in \mathcal{V}_1(x_l^*)$, $\nabla v(x_l^*)^T z = \nabla v(x_l^*)^T f(x_l^*)u = 0$. There must exist $\lambda_i^* \in \mathbb{R}$, $i = 1, \ldots, m$, such that $\nabla v(x_l^*) = \sum_{i=1}^m \lambda_i^* \nabla c_i(x_l^*)$. Otherwise $\exists \bar{z} \in \mathcal{V}_1(x_l^*)$, $\nabla v(x_l^*)^T \bar{z} \neq 0$. Therefore, $x_l^* \in \mathcal{S}$ is a KKT point [15, p. 328]. Furthermore, by Theorem 12.6 in [15, p. 345], x_l^* is a strict local minimum if $z^T \nabla_{xx} L(x_l^*, \lambda^*) z > 0$, for all $z \in \mathcal{V}_1(x_l^*)$, $z \neq 0$.

REFERENCES

[1] K. Tanabe, "A geometric method in nonlinear programming," *J. Optim. Theory Appl.*, vol. 30, no. 2, pp. 181–210, Feb. 1980.

[2] H. Yamashita, "A differential equation approach to nonlinear programming," *Math. Program.*, vol. 18, no. 1, pp. 155–168, Dec. 1980.

[3] A. A. Brown and M. C. Bartholomew-Biggs, "ODE versus SQP methods for constrained optimization," *J. Optim. Theory Appl.*, vol. 62, no. 3, pp. 371–386, Sep. 1989.

[4] S. Zhang and A. G. Constantinides, "Lagrange programming neural networks," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 39, no. 7, pp. 441–452, Jul. 1992.

[5] Z.-G. Hou, "A hierarchical optimization neural network for large-scale dynamic systems," *Automatica*, vol. 37, no. 12, pp. 1931–1940, Dec. 2001.

[6] L.-Z. Liao, H. Qi, and L. Qi, "Neurodynamical optimization," *J. Global Optim.*, vol. 28, no. 2, pp. 175–195, 2004.

[7] Y. Xia and J. Wang, "A recurrent neural network for solving nonlinear convex programs subject to linear constraints," *IEEE Trans. Neural Netw.*, vol. 16, no. 2, pp. 379–386, Mar. 2005.

[8] M. P. Barbarosou and N. G. Maratos, "A nonfeasible gradient projection recurrent neural network for equality-constrained optimization problems," *IEEE Trans. Neural Netw.*, vol. 19, no. 10, pp. 1665–1677, Oct. 2008.

[9] Y. Xia, G. Feng, and J. Wang, "A novel recurrent neural network for solving nonlinear optimization problems with inequality constraints," *IEEE Trans. Neural Netw.*, vol. 19, no. 8, pp. 1340–1353, Aug. 2008.

[10] S. Qin and X. Xue, "A two-layer recurrent neural network for nonsmooth convex optimization problems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 6, pp. 1149–1160, May 2015.

[11] P.-A. Absil, "Computation with continuous-time dynamical systems," in *Proc. Grand Challenge Non-Classical Comput. Int. Workshop*, York, U.K., Apr. 2005, pp. 18–19.

[12] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biological*, vol. 52, no. 3, pp. 141–152, Jul. 1985.

[13] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 3rd ed. Boston, MA, USA: Springer-Verlag, 2008.

[14] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Oper. Res.*, vol. 14, no. 4, pp. 699–719, Jul./Aug. 1966.

[15] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY, USA: Springer-Verlag, 1999.

[16] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[17] G. Chesi, A. Garulli, A. Tesi, and A. Vicino, "Solving quadratic distance problems: An LMI-based approach," *IEEE Trans. Autom. Control*, vol. 48, no. 2, pp. 200–212, Feb. 2003.

[18] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2003.

[19] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*. London, U.K.: Springer-Verlag, 1994.

[20] F. Morgan, *Real Analysis and Applications: Including Fourier Series and the Calculus of Variations*. Providence, RI, USA: AMS, 2005.

[21] A. Isidori, L. Marconi, and A. Serrani, *Robust Autonomous Guidance: An Internal Model-Based Approach*. London, U.K.: Springer-Verlag, 2003.

Quan Quan received the B.S. and Ph.D. degrees from Beihang University, Beijing, China, in 2004 and 2010, respectively.

He has been an Associate Professor with Beihang University since 2013. His current research interests include vision-based navigation and reliable flight control.

Kai-Yuan Cai received the B.S., M.S., and Ph.D. degrees from Beihang University, Beijing, China, in 1984, 1987, and 1991, respectively.

He has been a Full Professor with Beihang University since 1995. He is currently a Cheung Kong Scholar (Chair Professor), jointly appointed by the Ministry of Education of China and the Li Ka Shing Foundation of Hong Kong in 1999. His current research interests include software testing, software reliability, reliable flight control, and software cybernetics.