Article

TRANSACTIONS OF THE INSTITUTE OF MEASUREMENT & CONTROL

Transactions of the Institute of Measurement and Control
2019, Vol. 41(4) 1123–1138
© The Author(s) 2018
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0142331218798680
journals.sagepub.com/home/tim
SAGE

A non-linear estimation and model predictive control algorithm based on ant colony optimization

Hadi Nobahari and Saeed Nasrollahi

Abstract

A new heuristic controller, called the continuous ant colony controller, is proposed for non-linear stochastic Gaussian/non-Gaussian systems. The new controller formulates the state estimation and the model predictive control problems as a single stochastic dynamic optimization problem, and utilizes a colony of virtual ants to find and track the best estimated state and the best control signal. For this purpose, an augmented state space is defined. An integrated cost function is also defined to evaluate the points of the augmented state space, explored by the ants. This function minimizes simultaneously the state estimation error, tracking error, control effort and control smoothness. Ants search the augmented state space dynamically in a similar scheme to the optimization algorithm, known as the continuous ant colony system. Performance of the new model predictive controller is evaluated for three non-linear problems. The problems are a non-linear continuous stirred tank reactor, a non-linear cart and spring system, and the attitude control of a non-linear quadrotor. The results verify successful performance of the proposed algorithm from both estimation and control points of view.

Keywords

Ant colony optimization, heuristic control, model predictive control, multi-input multi-output control, non-linear stochastic system.

Introduction

Model predictive control (MPC) is an interesting topic in control engineering. The advantage of MPC approaches is their ability to handle dynamics with delays, constraints and non-linear elements (Camacho and Bordons, 2007; Maciejowski, 2002). MPC approaches have been employed in several practical examples, such as a quadrotor (Alexis et al., 2011, 2012), a pilot-scale distillation column (Huyck et al., 2014), a pneumatic muscle actuator (Huang et al., 2016), a four-degree-of-freedom robot (Best et al., 2015), and so on. Model predictive controllers are categorized as linear and non-linear ones, depending on the mathematical model of the system. Several linear model predictive controllers can be found within the literature, the most well known of which are model algorithmic control (MAC) (Richalet et al., 1978), dynamic matrix control (DMC) (Cutler and Ramaker, 1980), generalized predictive control (GPC) (Clarke et al., 1987), predictive functional control (PFC) (Richalet et al., 1987), and so on.

The domain of focus in this paper is non-linear MPC (NMPC). NMPC usually involves non-convex optimization problems. The authors categorize non-linear model predictive controllers to analytical and heuristic approaches. The most well known analytical approaches, for example extended dynamic matrix control (EDMC) (Haeri and Beik, 2005), multi-MPC (MMPC) (Özkan and Kothare, 2006), and a combination of feedback linearization and MPC (Van Soest et al., 2006), use non-linear and linear models for prediction. In comparison, there are NMPC approaches that utilize heuristic algorithms. Heuristic algorithms are able to solve non-linear optimization problems. Various heuristic algorithms have been developed to find the global minimum of non-linear functions, such as simulated annealing (SA) (Kirkpatrick et al., 1983), genetic algorithm (GA) (Goldberg and Holland, 1988), tabu search (TS) (Glover, 1989), particle swarm optimization (PSO) (Eberhart and Kennedy, 1995), ant colony optimization (ACO) (Dorigo et al., 1996), gravitational search algorithm (GSA) (Rashedi et al., 2009) and so on. Bagis (2011), Gozde and Taplamacioglu (2011), and Panda et al. (2012) used heuristic algorithms to tune the parameters of classical controllers. Furthermore, in Bououden et al. (2015), Huang et al. (2016), Júnior et al. (2014) and Smoczek and Szpytko (2014), ACO, PSO and evolutionary algorithms have been used to optimize the parameters of model predictive controllers. In this approach, the optimization is performed in an offline manner and the parameters are optimal only for the model used in the optimization process.

A control problem can be formulated as an estimation problem. In addition, an estimation problem can be formulated as a stochastic dynamic optimization problem. The combination of MPC and heuristic algorithms provides a powerful tool for computation of the approximate solutions

Department of Aerospace Engineering, Sharif University of Technology, Tehran, Iran

Corresponding author:
Hadi Nobahari, Department of Aerospace Engineering, Sharif University of Technology, Tehran, 1458889694, Iran.
Email: nobahari@sharif.edu

in control. In Salmond and Gordon (2001), a guidance problem has been modelled as a stochastic control problem. A particle-based control law has been used to solve the stochastic control problem. Here, a particle filter (PF) (Gordon et al., 1993) is used to optimize an inverse Gaussian cost function in a finite horizon. Najim et al. (2006) proposed a control algorithm based on sequential Monte Carlo (SMC). A Gaussian distribution is used to assign weights to the particles based on the average and the standard deviation of the costs. In Kantas et al. (2009), SMC algorithms have been used to find the optimal control sequence of stochastic MPC problems. Blackmore et al. (2010) presented a particle method for MPC of stochastic dynamic systems. The stochastic MPC problem is approximated as a deterministic one using a finite number of particles. The optimization problem is solved using mixed-integer linear programming techniques. De Villiers et al. (2011) used a Markov chain Monte Carlo (MCMC) algorithm to approximate the maximum *a posteriori* estimation of the control signal. In that study, PF has been used in the control computation step as a slight aid for the estimation of acceptance probability of an MCMC algorithm. In Pourjafari and Mojallali (2011), a modified discrete multi-valued PSO has been used to achieve a model predictive controller for voltage control system. Fan and Han (2012) proposed a non-linear model predictive controller based on the Gaussian particle swarm optimization. The proposed controller has been applied to a ball–plate system. Ho et al. (2012) presented a linear model predictive controller based on an ant colony algorithm. The ant colony algorithm has been used to find the optimal control signal for a blood glucose regulation problem. Sarailoo et al. (2015) proposed a non-linear model predictive controller based on a bee algorithm. The bee algorithm has been utilized in order to find the optimal control signal. The proposed controller has been applied to a three-tank system. A hybrid optimization algorithm has been proposed in Rajabi et al. (2016) by combining PSO with sequential quadratic programming (SQP) to find the optimal input of an NMPC problem. It has been assumed that all states are available. The proposed controller has been applied to an evaporator system.

In the above-mentioned references, particle-based heuristic methods are only used to compute the control signal. It is assumed that the state variables, required to compute the control signal are available and there is no need to estimate them, whereas in many applications, some states are not measured and must be estimated (Botchu and Ungarala, 2007). Furthermore, the above-mentioned references have not considered the dynamic effects of the estimator, in the closed-loop performance.

In Botchu and Ungarala (2007), the SMC filter in comparison with the moving horizon estimation (MHE) has been used for state estimation. Then, these estimated states are used in a normal model predictive controller. In Botchu and Ungarala (2007), it has been shown that the SMC filter is more accurate and faster than MHE. Therefore, it can help the convergence of the control signal and the real-time implementation of the NMPC. However, the stochastic systems with non-Gaussian noise have not been discussed. One approach that possibly comes closest to the current study is described in Stahl and Hauth (2011), in which two nested particle filters have been used – the first one estimating the states and the second estimating the optimal control signal of the stochastic NMPC problem. Consequently, two separate cost functions are defined. Calculation of the cost functions and the resampling process must be performed separately for each filter, which increases the computational time (Doucet and Johansen, 2009; Sileshi et al., 2013). In addition, in the second particle filter of Stahl and Hauth (2011), tuning of the weighting factors is difficult because the control smoothness and the tracking error, in the defined cost function, are not in the same order of magnitude and must be normalized, which is a difficult and boring task, especially in a non-affine system. However, in the algorithm proposed in the current study, the cost function terms are easily normalized with respect to the worst result, obtained by the ants. Moreover, in the current study, the state estimation and the control optimization tasks are performed by a single dynamic optimization module.

The contributions of the current study are summarized as follows: a multi-input multi-output (MIMO) heuristic controller, called the continuous ant colony controller (CACC), is proposed for non-linear stochastic Gaussian/non-Gaussian systems. Heuristic controllers need the current states of the system. The authors have integrated the state estimation and control problems as a single dynamic optimization problem, by defining a new augmented state space, which is an augmentation of the current state variables and the future control sequence. A group of virtual ants are exploring the augmented state space to estimate the states and to find the optimal sequence of the future control signal over a finite horizon. An integrated cost function is defined to evaluate the points, explored by the ants. This function simultaneously minimizes the state estimation error, tracking error, control effort and control smoothness. A dynamic algorithm based on ant colony optimization is developed to solve the integrated estimation and control problem. A single-input single-output variant of the proposed approach for non-linear stochastic Gaussian and stable systems was first presented in Nobahari and Nasrollahi (2016). The performance of the proposed approach is investigated in simulations using three non-linear stochastic systems. The results show that the new method can be used effectively for integrated state estimation and control optimization of a non-linear stochastic Gaussian/non-Gaussian system.

The rest of the paper is organized as follows: a non-linear control problem is defined in the next section. Then the standard MPC and a detailed description of the proposed controller are described. Numerical simulation and discussion are presented, and finally conclusions are drawn.

Problem definition

The problem is to estimate the states and to find the finite horizon optimal sequence of the control signal for a stochastic non-linear dynamic system. Evolution of the states and the outputs is modelled by the discrete time state space approach. Consider the following non-linear stochastic MIMO model:

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1), \boldsymbol{\omega}(k-1)) \tag{1}$$

$$\tilde{\mathbf{y}}(k) = \mathbf{h}(\mathbf{x}(k), \boldsymbol{v}(k)) \tag{2}$$

where k is the time step, $\mathbf{f} : \Re^{2n+s} \to \Re^n$ is a non-linear function of the states $\mathbf{x}(k-1) \in \Re^n$, the inputs $\mathbf{u}(k-1) \in \Re^s$ and the process noises $\boldsymbol{\omega}(k) \in \Re^n$, $\tilde{\mathbf{y}}(k) \in \Re^m$ contains the current measurements of the system and $\mathbf{h} : \Re^{n+p} \to \Re^m$ is a non-linear function of the states and the measurement noises $\boldsymbol{v}(k) \in \Re^p$. Furthermore, $\boldsymbol{\omega}(k)$ and $\boldsymbol{v}(k)$ can be Gaussian or non-Gaussian.

Model predictive control

MPC (Richalet et al., 1987) is a powerful approach for optimal control of linear and non-linear systems. The advantage of MPC is its ability to handle non-linear systems and multivariable systems with known delays and constraints (Camacho and Bordons, 2007; Maciejowski, 2002). The nominal discrete-time state space representation of a non-linear system is formulated as:

$$\mathbf{x}(k) = \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1)) \tag{3}$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k)) \tag{4}$$

where $\mathbf{f} : \Re^{n+s} \to \Re^n$ is a non-linear function of the states and the inputs, $\mathbf{y}(k) \in \Re^m$ contains the current outputs of the system and $\mathbf{h} : \Re^n \to \Re^m$ is a non-linear function of the states. In the MPC approach, the dynamic model is used to predict the future outputs at every time step k for a certain prediction horizon $\mathrm{T_P}$. An optimization algorithm calculates the optimal sequence of the future control signal over a prediction horizon according to the cost function and the constraints. Then, the first value of the optimal control signal, $\mathbf{u}(k)$, is applied to the system as the current input. The optimization procedure is repeated when a new measurement is available. Optimal control sequence is obtained by minimizing the standard MPC cost function, defined as (Camacho and Bordons, 2007)

$$\begin{aligned} J = &\sum_{i=k+1}^{k+\mathrm{T_p}} (\mathbf{x}_\mathrm{d}(i) - \mathbf{x}(i))^\mathrm{T} \mathbf{Q} (\mathbf{x}_\mathrm{d}(i) - \mathbf{x}(i)) \\ &+ \sum_{i=k}^{k+\mathrm{T_p}-1} (\mathbf{u}(i) - \mathbf{u}(i-1))^\mathrm{T} \mathbf{R} (\mathbf{u}(i) - \mathbf{u}(i-1)) \end{aligned} \tag{5}$$

where $\mathbf{x}_\mathrm{d}$ is the set point, and $\mathbf{Q}$ and $\mathbf{R}$ are appropriate weighting matrices.

Continuous ant colony controller

The aim is to develop an algorithm based on a continuous ant colony system (CACS) (Pourtakdoust and Nobahari, 2004) to estimate the current states and the optimal sequence of the future controls, simultaneously. For this purpose, an augmented state vector is defined as follows:

$$\boldsymbol{\chi}(k) \equiv \begin{bmatrix} \mathbf{x}(k) \\ u_1(k) \\ u_1(k+1) \\ \vdots \\ u_1(k+\mathrm{T_p}-1) \\ \vdots \\ u_s(k) \\ u_s(k+1) \\ \vdots \\ u_s(k+\mathrm{T_p}-1) \end{bmatrix} \tag{6}$$

Now, using Equations (1) and (2), and assuming $u(k) = u(k-1) + v(k-1)$, as proposed in Stahl and Hauth (2011), the evolution model of the problem can be rewritten as follows:

$$\begin{aligned} \boldsymbol{\chi}(k) &= \mathbf{f}_\mathrm{a}(\mathbf{x}(k-1), \mathbf{u}(k-1), \boldsymbol{\omega}(k-1), \mathbf{v}_i(k-1)) \\ &\equiv \begin{bmatrix} \mathbf{f}(\mathbf{x}(k-1), \mathbf{u}(k-1), \boldsymbol{\omega}(k-1)) \\ u_1(k-1) + v_1(k-1) \\ u_1(k) + v_1(k) \\ \vdots \\ u_1(k+\mathrm{T_p}-2) + v_1(k+\mathrm{T_p}-2) \\ \vdots \\ u_s(k-1) + v_s(k-1) \\ u_s(k) + v_s(k) \\ \vdots \\ u_s(k+\mathrm{T_p}-2) + v_s(k+\mathrm{T_p}-2) \end{bmatrix} \end{aligned} \tag{7}$$

$$\tilde{\mathbf{y}}(k) = \mathbf{h}(\mathbf{x}(k), \boldsymbol{v}(k)) \tag{8}$$

where s is the number of inputs, $\mathrm{T_P}$ is the prediction horizon, $\boldsymbol{\chi}(k) \in \Re^{\mathrm{n}+s\mathrm{T_p}}$ is the augmented state vector, $\mathbf{f}_\mathrm{a} : \Re^{2n+(1+2\mathrm{T_P})s} \to \Re^{n+s\mathrm{T_P}}$ is the augmented state space, $\mathbf{u}(k) = [u_1(k) \ \cdots \ u_s(k)]^\mathrm{T}$ contains the current inputs, $\mathbf{v}_i(k-1) \in \Re^{\mathrm{T_p}}$, defined as $\mathbf{v}_i(k-1) = [v_i(k-1) \ \cdots \ v_i(k+\mathrm{T_p}-2)]^\mathrm{T}$, $(i = 1, \ldots, s)$, is a vector of zero mean Gaussian random variables with standard deviation σ_v. Another definition of an augmented state vector is found in some previous works. For example, in Yin et al. (2017a, 2017b), unknown disturbances are augmented with the states, and a sliding mode observer is used for simultaneous estimation of the states and the disturbances for switched systems.

The CACC is a new heuristic controller, proposed for non-linear stochastic systems. Figure 1 shows the flowchart of CACC. The augmented state vector, $\boldsymbol{\chi}(k)$, defined in Equation (6), is utilized to integrate the state estimation and the finite horizon optimal control problems. CACC has two loops: a main outer loop, iterating every time a new measurement is arrived, and an inner loop, iterating to find the best estimation of the current states and the best optimal controls for the future. At first, virtual ants are distributed randomly within the augmented state space. Then, the position of ants

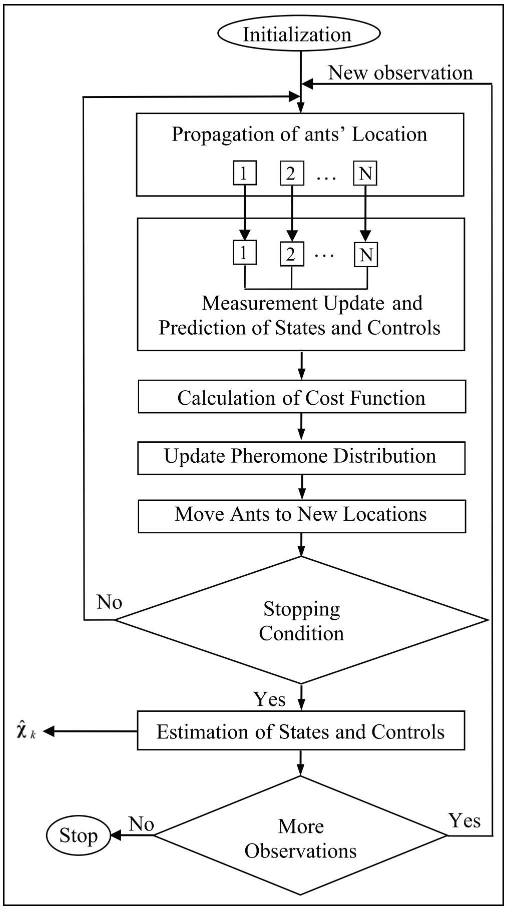

Initialization
New observation
Propagation of ants' Location
1
2
…
N
1
2
…
N
Measurement Update and Prediction of States and Controls
Calculation of Cost Function
Update Pheromone Distribution
Move Ants to New Locations
No
Stopping Condition
Yes
$\hat{\chi}_k$
Estimation of States and Controls
Stop
No
More Observations
Yes

Figure 1. Flowchart of continuous ant colony controller.

is propagated using the initial position of them. Next, the expected measurement corresponding to each ant is calculated. Then, the future states and controls are predicted, in a finite horizon, using the augmented state space model, and the integrated cost of estimation and control is calculated for each ant. Ants use their experience to update the pheromone distribution over the augmented state space. The inner loop is terminated when a predefined number of iterations are performed. Finally, the current states and the future controls are estimated using a mean operator. More details will be described in later subsections.

Initialization

CACC has some control parameters. These parameters are set at the beginning. Moreover, the initial value of the states and the initial value of the future control sequence ($\mathbf{x}_{l,j}(k)$ and $\mathbf{u}_{l,j}(k)$, for $j = 1, \ldots, \mathrm{N}$ and $k = 0$) are set, where N is the number of ants and l is the number of iterations. A random generator with uniform distribution is used to generate $\mathbf{x}_{l,j}(0)$ and $\mathbf{u}_{l,j}(0)$ within the acceptable range of the states and controls.

Propagation, prediction and measurement update

At the lth iteration of the inner loop, the augmented state vector of ant j at time step $k-1$ $\boldsymbol{\chi}_{l,j}(k-1)$, defined previously, is propagated as

$$\boldsymbol{\chi}_{l,j}(k) = \mathbf{f}_{\mathrm{a}}\big(\mathbf{x}_{l,j}(k-1), \mathbf{u}_{l,j}(k-1), \boldsymbol{\omega}_{l,j}(k-1), \mathbf{v}_{l,j}(k-1)\big) \quad (9)$$

For this purpose the control signal at time step $k-1$, is propagated for ant j as

$$\mathbf{u}_{l,j}(k+i-1) = \mathbf{u}_{l,j}(k+i-2) + \mathbf{v}_{l,j}(k+i-2), \quad (i = 1, \ldots, \mathrm{T_p}) \quad (10)$$

It should be noted that initially $\mathbf{u}_{l,j}(k-1) = \hat{\mathbf{u}}(k-1)$. Then, the future states of ant j at iteration l, is predicted as

$$\mathbf{x}_{l,j}(k+\mathrm{i}-1) = \mathbf{f}\big(\mathbf{x}_{l,j}(k+i-2), \mathbf{u}_{l,j}(k+i-2), \boldsymbol{\omega}_{l,j}(k+i-2)\big) \quad (i = 1, \ldots, \mathrm{T_p}) \quad (11)$$

where $\boldsymbol{\omega}_j$ and $\mathbf{v}_j$ have been defined earlier. Furthermore, the measurement at time step k and iteration l, expected by ant j, is calculated as follows:

$$\mathbf{y}_{l,j}(k) = \mathbf{h}\big(\mathbf{x}_{l,j}(k), 0\big) \quad (12)$$

Calculation of the cost function

Each ant is assigned a cost, based on the quality of its current position within the augmented state space. The cost, assigned in iteration l to ant j at time step k, denoted by $J_{l,j}(k)$, is calculated as

$$J_{l,j}(k) = \mathrm{w_{EE}}J_{\mathrm{EE}} + \mathrm{w_{TE}}J_{\mathrm{TE}} + \mathrm{w_{CE}}J_{\mathrm{CE}} + \mathrm{w_{CS}}J_{\mathrm{CS}} \quad (13)$$

where

$$J_{\mathrm{EE}} = \frac{[\tilde{\mathbf{y}}(k) - \mathbf{y}_{l,j}(k)]^{\mathrm{T}}[\tilde{\mathbf{y}}(k) - \mathbf{y}_{l,j}(k)]}{\max\limits_{j=1}^{\mathrm{N}} [\tilde{\mathbf{y}}(k) - \mathbf{y}_{l,j}(k)]^{\mathrm{T}}[\tilde{\mathbf{y}}(k) - \mathbf{y}_{l,j}(k)]} \quad (14)$$

$$J_{\mathrm{TE}} = \frac{\sum\limits_{i=k+1}^{k+\mathrm{T_p}} [\mathbf{x}_{\mathrm{d}}(i) - \mathbf{x}_{l,j}(i)]^{\mathrm{T}}[\mathbf{x}_{\mathrm{d}}(i) - \mathbf{x}_{l,j}(i)]}{\max\limits_{j=1}^{\mathrm{N}} \sum\limits_{i=k+1}^{k+\mathrm{T_p}} [\mathbf{x}_{\mathrm{d}}(i) - \mathbf{x}_{l,j}(i)]^{\mathrm{T}}[\mathbf{x}_{\mathrm{d}}(i) - \mathbf{x}_{l,j}(i)]} \quad (15)$$

$$J_{\mathrm{CE}} = \frac{\sum\limits_{i=k}^{k+\mathrm{T_p}-1} [\mathbf{u}_{l,j}(i)]^{\mathrm{T}}[\mathbf{u}_{l,j}(i)]}{\max\limits_{j=1}^{\mathrm{N}} \sum\limits_{i=k}^{k+\mathrm{T_p}-1} [\mathbf{u}_{l,j}(i)]^{\mathrm{T}}[\mathbf{u}_{l,j}(i)]} \quad (16)$$

$$J_{\mathrm{CS}} = \frac{\sum\limits_{i=k}^{k+\mathrm{T_p}-1} [\Delta\mathbf{u}_{l,j}(i)]^{\mathrm{T}}[\Delta\mathbf{u}_{l,j}(i)]}{\max\limits_{j=1}^{\mathrm{N}} \sum\limits_{i=k}^{k+\mathrm{T_p}-1} [\Delta\mathbf{u}_{l,j}(i)]^{\mathrm{T}}[\Delta\mathbf{u}_{l,j}(i)]} \quad (17)$$

In the above equations, to minimize the estimation error, J_{EE} penalizes the innovation (the difference between the real measurement, $\tilde{\mathbf{y}}(k)$, and the calculated measurement, $\mathbf{y}_{l,j}(k)$, expected by ant j), normalized with respect to the worst result, obtained by the ants; J_{TE} is the normalized penalty, considered for the tracking error (difference between the future states, $\mathbf{x}_{l,j}(i), (i = k+1, \ldots, k+\mathrm{T_P})$, expected by ant j, and the set point, $\mathbf{x}_{\mathrm{d}}(i)$); J_{CE} penalizes the control effort over a certain prediction horizon; J_{CS} penalizes the control smoothness through minimizing the control fluctuation defined as $\Delta\mathbf{u}_{l,j}(k) = \mathbf{u}_{l,j}(k) - \mathbf{u}_{l,j}(k-1)$. In addition, $\mathrm{w_{EE}}$, $\mathrm{w_{TE}}$, $\mathrm{w_{CE}}$ and $\mathrm{w_{CS}}$ are constant weighting factors that control the relative importance of the cost terms. These parameters are tuned in a trial-and-error procedure. The tuned values will be presented below. In this way, different points of the augmented state space are evaluated and some knowledge about the problem is acquired, which is used to update the pheromone distribution.

Updating pheromone distribution

CACC utilizes the same pheromone model and pheromone updating rule as in CACS. A multi-variable Gaussian function is used to model the pheromone distribution over the augmented state space. In every iteration of the inner loop, pheromone distribution is updated using the acquired knowledge by the ants. The cost of the new points, explored by the ants, is calculated using Equation (13). The best point is assigned to $\boldsymbol{\chi}_{l,\min}(k)$. In addition, variance of the Gaussian pheromone distribution is updated based on the cost of the evaluated points and the aggregation of those points around the best one. To satisfy simultaneously the fitness and aggregation criteria, the weighted variance, proposed in Pourtakdoust and Nobahari (2004), is utilized to update the pheromone distribution function in dimension d of the augmented state space as follows:

$$[\sigma_l^d(k)]^2 = \frac{\sum_{j=1}^{\mathrm{N}} \frac{1}{J_{l,j}(k) - J_{l,\,\min}(k)} [\chi_{l,\mathrm{j}}^d(k) - \chi_{l,\min}^d(k)]^2}{\sum_{j=1}^{\mathrm{N}} \frac{1}{J_{l,j}(k) - J_{l,\,\min}(k)}} \tag{18}$$

This strategy means that the centre of the region, discovered during the subsequent iteration, is the last best point and the narrowness of its width depends on the aggregation of other competitors around the best point. During each iteration, the height of pheromone distribution increases with respect to the previous iteration and its narrowness decreases (Pourtakdoust and Nobahari, 2004).

Movement of the ants

In any iteration, ants move from their current position to their destination using the current pheromone distribution, modelled as a normal function. The centre of the normal distribution function in dimension d is the best point $\chi_{l,\min}^d(k)$, found from the first iteration and its variance is $[\sigma_l^d(k)]^2$. Normal distribution permits all points of the continuous augmented state space to be selected, either close to or far from the best one.

Stopping condition

CACC has two loops. The inner loop stops when the maximum number of iteration ($l_{\max}$) is reached. The outer loop stops when the measurements are finished.

Estimation of the states and controls

After termination of the inner loop, ants are ranked based on their cost and the augmented state vector at time step k is estimated based on the average position of top ants:

$$\hat{\boldsymbol{\chi}}(k) = \frac{1}{\mathrm{N_t}} \sum_{j=1}^{\mathrm{N_t}} \hat{\boldsymbol{\chi}}_{l_{\max},j}(k) \tag{19}$$

where $\mathrm{N_t} < \mathrm{N}$ is a parameter of the algorithm and denotes the number of top ants. The estimated vector in augmented state space consists of the estimated states and the estimated control sequence. Therefore, the estimated states (elements 1, ..., n of $\hat{\boldsymbol{\chi}}$) are calculated using the estimated states of $\mathrm{N_t}$ top ants (elements 1, ..., n of $\hat{\boldsymbol{\chi}}_{l_{\max},j}, j = 1, \ldots, \mathrm{N_t}$) and the estimated controls for time step k (elements n+1, n+$\mathrm{T_p}$, ..., n+$s\mathrm{T_p}$ of $\hat{\boldsymbol{\chi}}$) are calculated using the estimated controls of $\mathrm{N_t}$ top ants (elements n+1, n+$\mathrm{T_p}$, ..., n+$s\mathrm{T_p}$ of $\hat{\boldsymbol{\chi}}_{l_{\max},j}$, $j = 1, \ldots, \mathrm{N_t}$).

Numerical simulation and discussion

In this section, CACC is applied to three non-linear systems. The first problem is a non-linear continuous stirred tank reactor (CSTR), which is a non-affine system; the second problem is a non-linear cart and spring system; and the third one is the attitude control of a quadrotor. Results are given in subsequent sections.

A continuous stirred tank reactor

A CSTR process (Botchu and Ungarala, 2007; Stahl and Hauth, 2011) is an irreversible exothermic chemical reaction (Figure 2). Here, $\mathrm{C_f}$ and $\mathrm{T_0}$ represent the feed concentration and the feed temperature, respectively. The coolant stream, $q_c(t)$, is the control input of the system. The current temperature of the system, $T(t)$, is measured and the objective is to control the concentration, $C(t)$, which is not measured directly and must be estimated. Table 1 represents the parameters of the CSTR. The stochastic differential equations of this system are stated as follows (Stahl and Hauth, 2011):

$$\dot{C}(t) = \frac{\mathrm{q}}{\mathrm{V}} (\mathrm{C_f} - C(t)) - \mathrm{k_0}\, C(t)\, e^{\frac{-\mathrm{E}}{\mathrm{R}\, T(t)}} + \omega(t) \tag{20}$$

$$\begin{aligned} \dot{T}(t) = {} & \frac{\mathrm{q}}{\mathrm{V}} (\mathrm{T_0} - T(t)) - \frac{\mathrm{k_0}\, \Delta\mathrm{H}}{\rho\, \mathrm{C_p}} C(t) \mathrm{e}^{\frac{-\mathrm{E}}{\mathrm{R}\, T(t)}} \\ & + \frac{\rho_\mathrm{c}\, \mathrm{C_{pc}}}{\rho\, \mathrm{C_p} \mathrm{V}}\, q_c(t) \left(1 - \mathrm{e}^{\frac{-\mathrm{hA}}{\rho_\mathrm{c}\, \mathrm{C_{pc}}\, q_c(t)}}\right) (\mathrm{T_{co}} - T(t)) \end{aligned} \tag{21}$$

$$y(t) = T(t) + \upsilon(t) \tag{22}$$

where $\omega(\mathrm{t}) \sim \mathrm{N}(0, \sigma_\omega^2)$ and $\upsilon(\mathrm{t}) \sim \mathrm{N}(0, \sigma_\upsilon^2)$ are the process and measurement noises, respectively. They are continuous zero mean white noises with known covariance $\mathrm{E}\{\omega(t)\omega(t+\tau)\}$

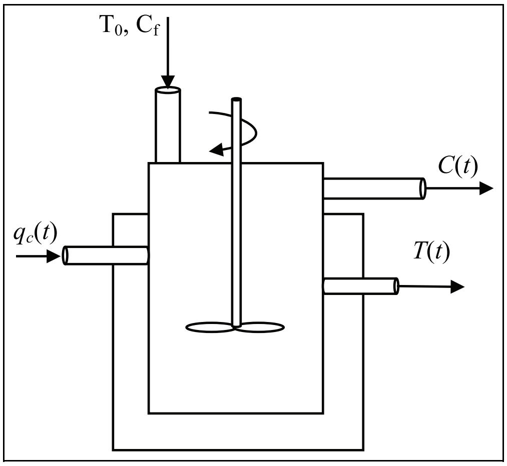


Figure 2. Schematic of a continuous stirred tank reactor (CSTR) system (Stahl and Hauth, 2011).

Table 1. Constant parameters of continuous stirred tank reactor (CSTR) problem (Stahl and Hauth, 2011).

Parameter	Value	Unit	Description
C_p, C_{pc}	1	kcal/g	Specific heat coefficients
ρ_c, ρ	1000	g/l	Liquid densities
C_f	1	mol/l	Feed concentration
ΔH	-2×10^5	cal/mol	Heat of reaction
hA	7×10^5	kcal/min	Heat transfer term
q	100	l/min	Process flow rate
k_0	7.2×10^{10}	l/min	Reaction rate constant
T_0	350	K	Feed temperature
T_{co}	350	K	Inlet coolant temperature
E/R	1×10^4	K	Activation energy term
V	100	l	Reactor volume

$= Q_1\delta(\tau)$ and $E\{v(t)v(t+\tau)\} = Q_2\delta(\tau)$, where $Q_i(i = 1, 2)$ is the corresponding power spectral density and $\delta(.)$ is the Dirac delta function (Varziri et al., 2008). To discretize the stochastic differential equations, the Euler–Maruyama method is used (Kloeden and Platen, 1992). The sampling time is 0.05 min, as in Stahl and Hauth (2011), and other parameters needed for simulation are set as $\sigma_\omega = 10^{-6}$, $\sigma_v = 0.5$, C(0) = 0.06 mol/l, T(0) = 449 K, and $q_c(0) = 102.411$ 1/min.

The performance of CACC is evaluated through numerical simulation. The parameters of CACC are listed in Table 2. These parameters are tuned in a trial-and-error procedure. The values of w_{EE}, w_{TE}, w_{CE} and w_{CS} depend on the relative importance of the corresponding performance indices. The length of the prediction horizon affects the overshoot of the system (Rossiter, 2013). Furthermore, the maximum value of σ_v depends on the lower and upper bounds of the control signal (Stahl and Hauth, 2011). The number of ants and the maximum number of iterations control the computational cost and the performance of the algorithm.

Figure 3(a) compares the true and estimated concentration. Figure 3(b) shows the control signal. The estimated, true and measured temperature are compared in Figure 3(c). CACC is also applied to the CSTR problem when a random generator with uniform distribution is used to generate the process and measurement noises. The results are shown in Figure 4. It can be observed that the proposed approach can simultaneously estimate the states and find the optimal control of CSTR with both Gaussian and non-Gaussian noise distribution.

To investigate the effect of the parameters of the proposed controller, a sensitivity analysis is performed. The results are depicted in Figure 5. Figures 5(a) and (b) show the effect of w_{TE} on the control performance and the control signal. Decreasing the value of w_{TE} decreases overshoot and increasing its value increases the control effort. Figures 5(c) and (d) show that decreasing the value of w_{CE} increases the control effort. The effect of w_{CS} is illustrated in Figures 5(e) and (f). Decreasing the value of w_{CS} increases the control effort. Figures 5(g) and (h) show that decreasing the value of T_p increases overshoot and control effort. Figures 5(i) and (j) show the effect of σ_v. Decreasing the value of σ_v increases rise time and increasing its value increases the control effort. Decreasing the value of w_{EE} increases rise time and overshoot. Finally, increasing the value of N_t increases the control effort and decreasing its value increases rise time.

Now, a Monte Carlo (Rubinstein and Kroese, 2016) simulation is performed to evaluate the statistical performance of CACC in solving CSTR problem. For this purpose, the root mean square error (RMSE) is calculated. Here, error means the average error of the estimated states. The RMSE versus the number of runs is illustrated in Figure 6. It shows that 100 runs are enough to evaluate the statistical performance of CACC.

Table 2. Parameters of continuous ant colony controller (CACC).

Problem	Parameters							
	w_{EE}	w_{TE}	w_{CE}	w_{CS}	T_p	σ_v	N	I_{max}
CSTR	10	5	1	0.1	10	0.01	20	10
Cart and spring	1	20	2	0.1	4	0.1	25	10
Quadrotor problem (case 1)	0	100	1	0.1	8	0.01	30	20
Quadrotor problem (case 2)	10	100	1	0.1	5	0.01	20	15

CSTR, continuous stirred tank reactor.

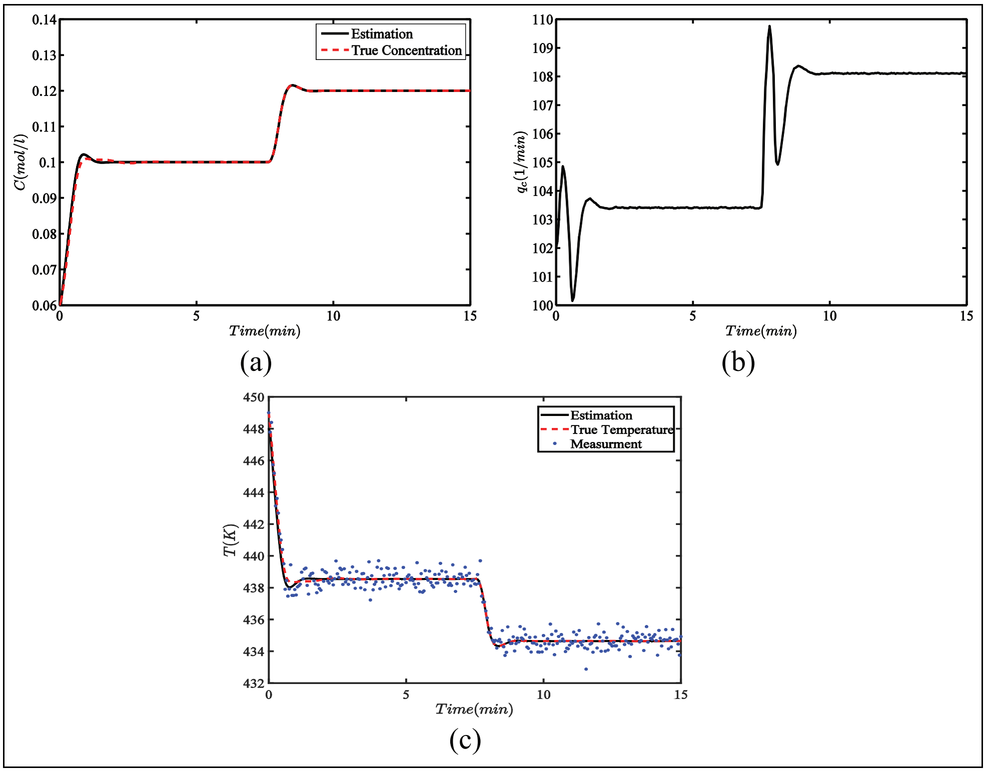


Figure 3. Performance of the proposed algorithm in continuous stirred tank reactor (CSTR) problem (with Gaussian noise): (a) estimated and true concentration; (b) control signal; (c) estimated, true, and measured temperature.

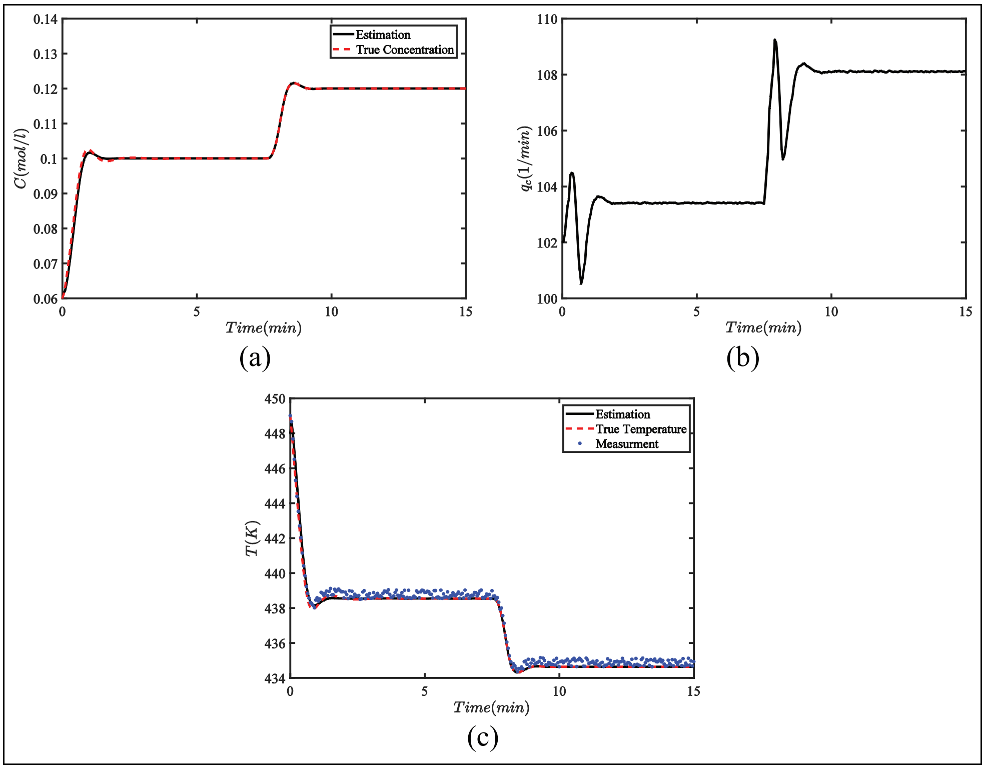
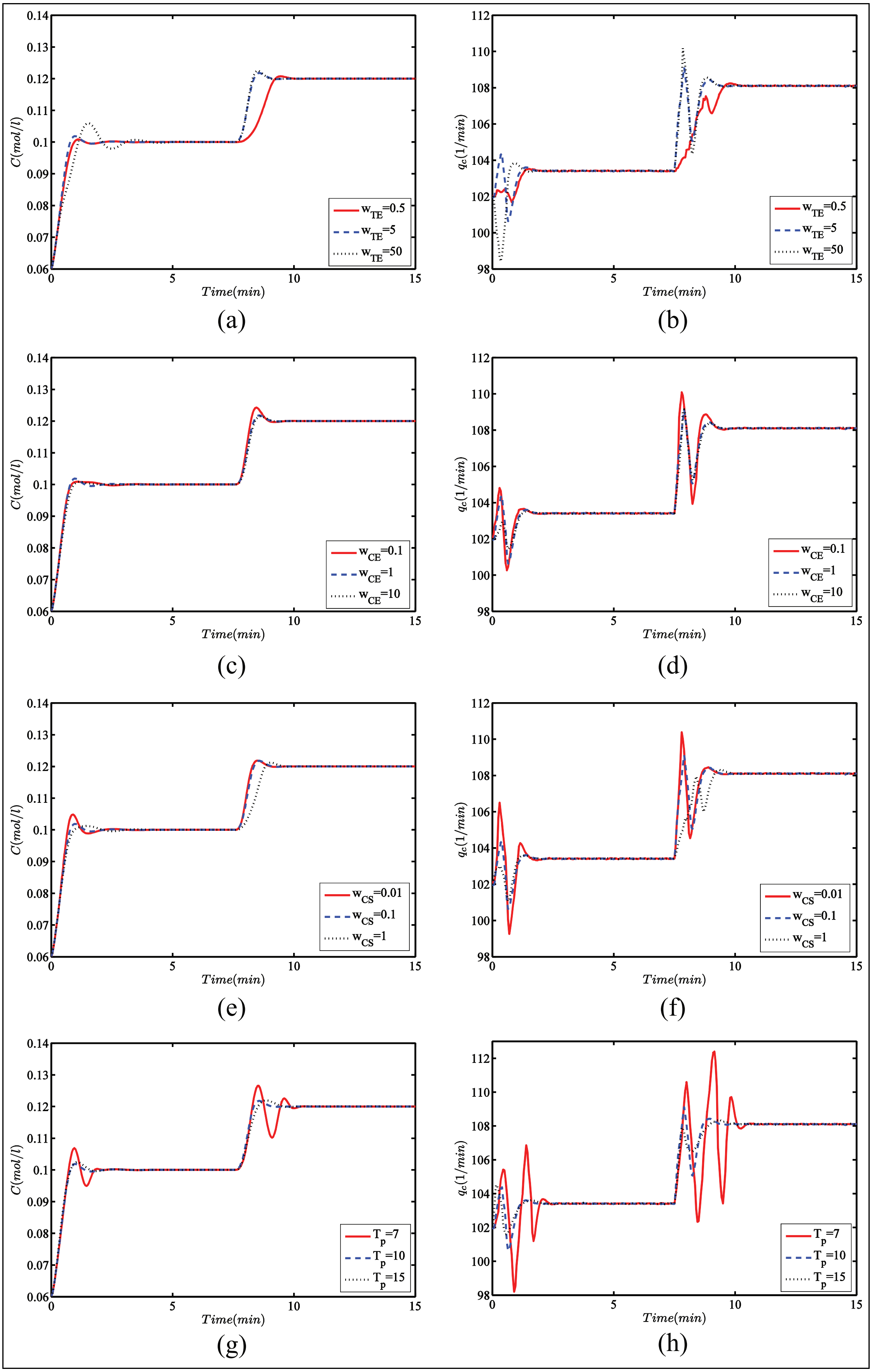


Figure 4. Performance of the proposed algorithm in continuous stirred tank reactor (CSTR) problem (with non-Gaussian noise): (a) estimated and true concentration; (b) control signal; (c) estimated, true, and measured temperature.



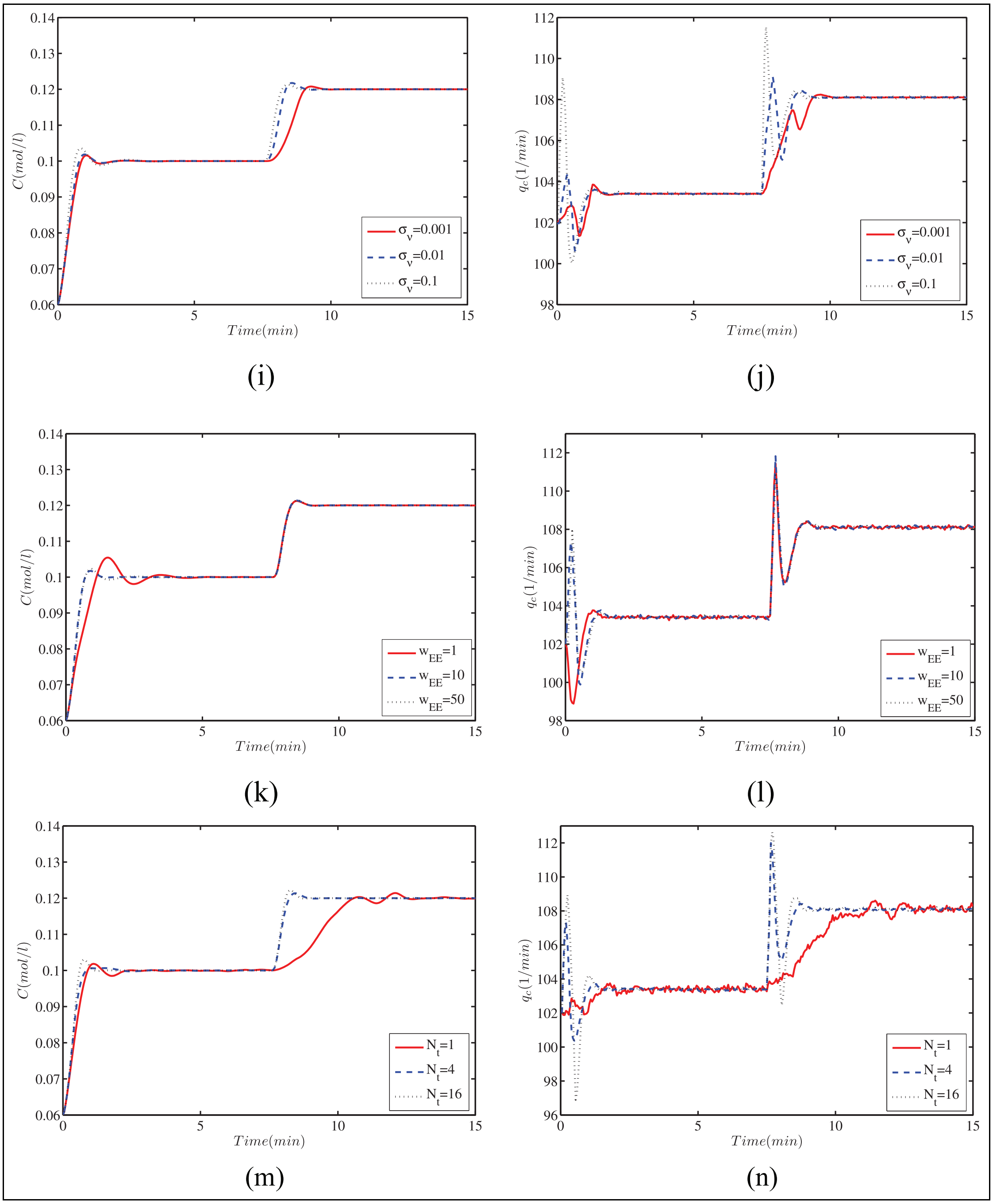
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 5. Continued



(i) (j)

(k) (l)

(m) (n)

Figure 5. Sensitivity analysis of CACC performance to its parameters in solving continuous stirred tank reactor (CSTR) problem: (a) and (b) the effect of w_{TE} on the control performance and control signal; (c) and (d) effect of w_{CE}; (e) and (f) effect of w_{CS}; (g) and (h) effect of T_p; (i) and (j) effect of σ_v; (k) and (l) effect of w_{EE}; (m) and (n) effect of N_t.

The numerical results and the Monte Carlo simulation verify the convergence of the proposed algorithm. In Stutzle and Dorigo (2002), a convergence proof for a wide class of ACO algorithms has been presented. The presented theorem states that ACO finds an optimal solution with arbitrarily large probability if run long enough (Stutzle and Dorigo, 2002). In addition, the convergence of the iterative methods has been discussed in Nemati and Yousefi (2016a, 2016b).

Non-linear cart and spring system

The second problem is a non-linear cart and spring system (Rubagotti et al., 2011) taking into account the effect of elastic spring. The objective is to regulate the displacement of the carriage from the equilibrium position (Figure 7), when only the velocity of the carriage is measured. The states $x_1(t)$ and $x_2(t)$ represent the position and velocity of the carriage, $u(t)$ is

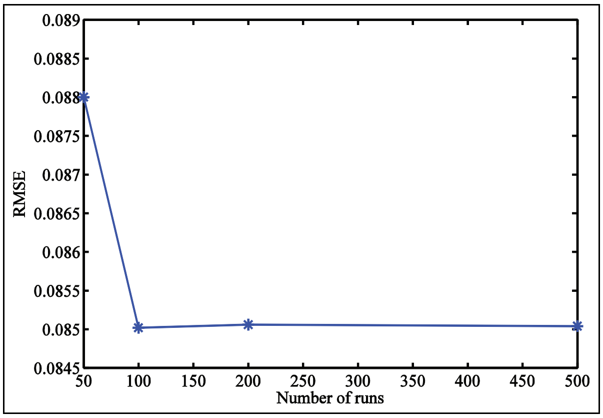


Figure 6. Root mean square error (RMSE) versus the number of runs in solving continuous stirred tank reactor (CSTR) problem.

the system input and $K(t) = K_0 e^{-x_1(t)}$ is the non-linear stiffness of the spring. The numerical value of the system parameters are $M = 1$ kg, $K_0 = 0.33$ N/m and $h_d = 1.1$ Ns/m, as in Rubagotti et al. (2011).

The non-linear differential equations are stated as (Grancharova and Johansen, 2012; Rubagotti et al., 2011)

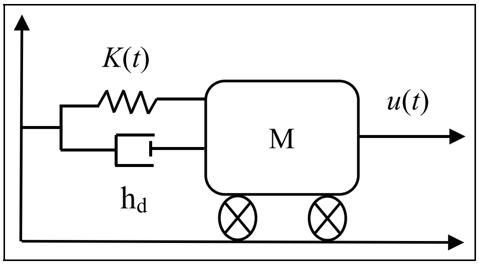


Figure 7. Schematic of a cart and spring system (Rubagotti et al., 2011).

$$\dot{x}_1(t) = x_2(t) + \omega(t) \tag{23}$$

$$\dot{x}_2(t) = -\frac{K_0}{M} e^{-x_1(t)} x_1(t) - \frac{h_d}{M} x_2(t) + \frac{u(t)}{M} \tag{24}$$

$$y(t) = x_2(t) + \upsilon(t) \tag{25}$$

The noises $\omega(t) \sim N(0, \sigma_\omega^2)$ and $\upsilon(t) \sim N(0, \sigma_\upsilon^2)$ are white, uncorrelated and have known variances. Similar to the previous example, the Euler–Maruyama method is used to discretize the system. Here, the sampling time is 0.4 s, as in Rubagotti et al. (2011), and other parameters of the system, needed for simulation, are set as $x_1(0) = 2$m, $x_2(0) = 1.7$m/s, $u(0) = -1.6$N, $\sigma_\omega = 0.001$ and $\sigma_\upsilon = 0.8$. Time response of the states and the control signal are depicted in Figure 8. Figure 9(a) illustrates the estimated and true position of the carriage. The estimated, true and measured outputs are represented in Figure 9(b). Results show that CACC can properly estimate the states and control the position.

Attitude control of a quadrotor

Another problem to evaluate the proposed controller is the attitude control of a quadrotor, which is a MIMO problem. Figure 10 shows the configuration of a quadrotor from the top view. The *i*th rotor rotates around the vertical axis of the body frame (z_B) with angular velocity Ω_i ($i = 1, \ldots, 4$). Rotors 1 and 3 rotate clockwise, and rotors 2 and 4 rotate counter clockwise. The three degrees-of-freedom (3-DOF) equations used for the attitude control problem are stated as follows (Choi and Ahn, 2015; Modirrousta and Khodabandeh, 2015; Nobahari and Sharifi, 2014):

$$\dot{x}_1(t) = x_2(t) \tag{26}$$

$$\dot{x}_2(t) = \frac{(I_{yy} - I_{zz})}{I_{xx}} x_4(t) x_6(t) + \frac{u_1(t)}{I_{xx}} + \frac{I_{rotor}}{I_{xx}} \Omega_r x_4(t) \tag{27}$$

$$\dot{x}_3(t) = x_4(t) \tag{28}$$

$$\dot{x}_4(t) = \frac{(I_{zz} - I_{xx})}{I_{yy}} x_2(t) x_6(t) + \frac{u_2(t)}{I_{yy}} - \frac{I_{rotor}}{I_{yy}} \Omega_r x_4(t) \tag{29}$$

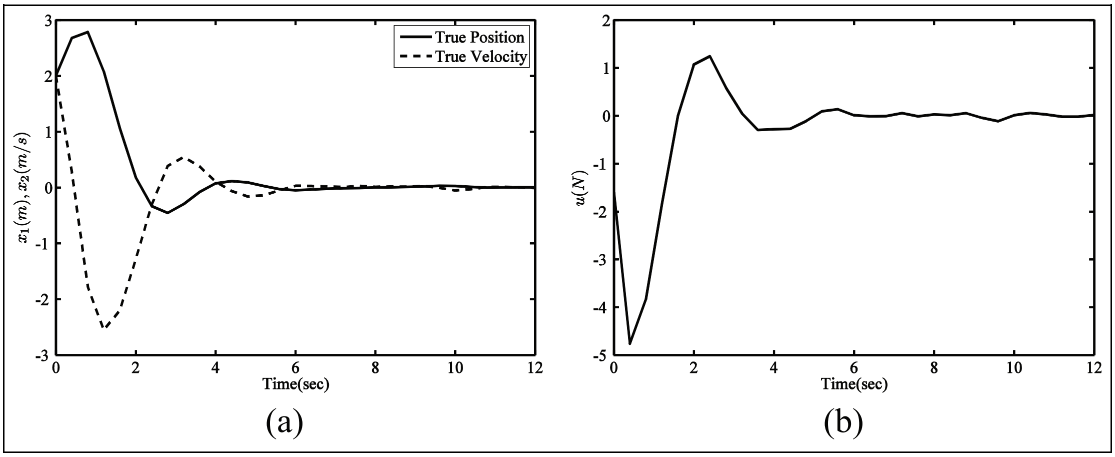


Figure 8. Performance of the controller: (a) state variables; (b) control signal.

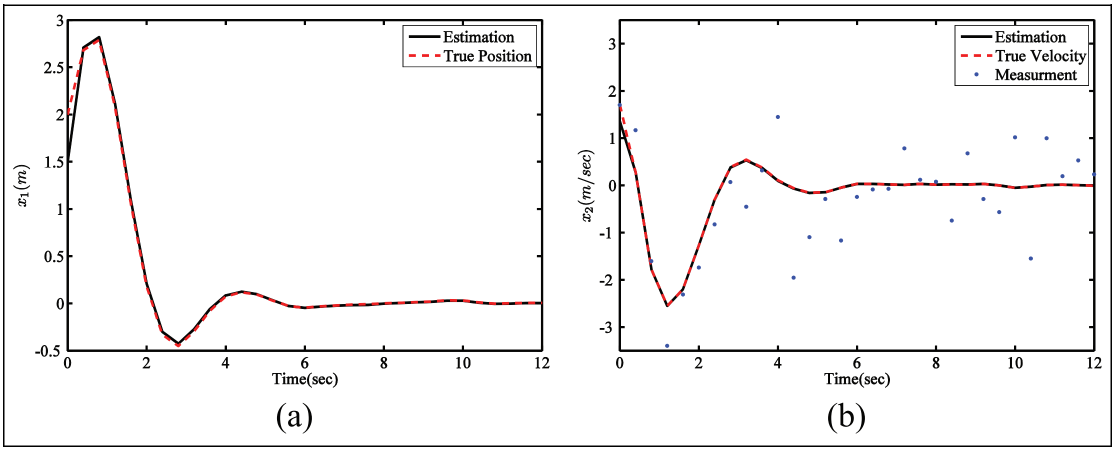


Figure 9. Performance of the estimation: (a) comparison of the estimated and the true position; (b) comparison of the estimated; true and measured velocity.

$$\dot{x}_5(t) = x_6(t) \tag{30}$$

$$\dot{x}_6(t) = \frac{(I_{xx} - I_{yy})}{I_{zz}} x_2(t) x_4(t) + \frac{u_3(t)}{I_{zz}} \tag{31}$$

where $x_1 = \phi$, $x_3 = \theta$ and $x_5 = \psi$ are roll, pitch and yaw angles, and $x_2 = \dot{\phi}$, $x_4 = \dot{\theta}$ and $x_6 = \dot{\psi}$ are the angular rates. Moreover, I_{xx}, I_{yy} and I_{zz} are the quadrotor moments of inertia, and I_{rotor} is the moment of inertia of a rotor about its axis. Furthermore, Ω_r is defined as

$$\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \tag{32}$$

Finally, u_1, u_2, u_3 and T_t are roll, pitch and yaw moments, and the total thrust, stated as

$$u_1(t) = bd_{cg}(\Omega_2^2 - \Omega_4^2) \tag{33}$$

$$\mathrm{u}_2(t) = bd_{cg}(\Omega_1^2 - \Omega_3^2) \tag{34}$$

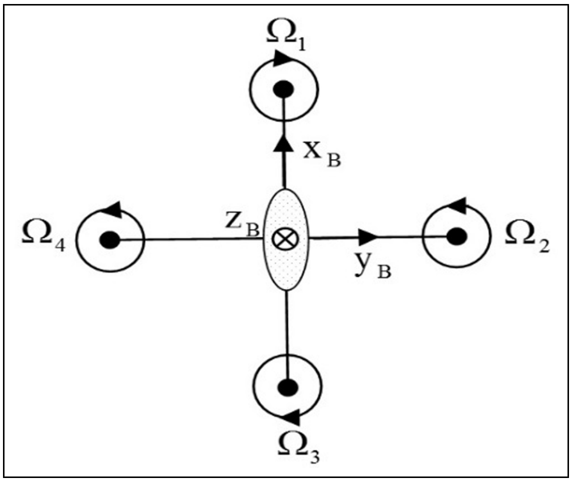


Figure 10. Configuration of a quadrotor from the top view.

$$u_3(t) = d(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \tag{35}$$

$$T_t = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{36}$$

where b, d and d_{cg} are thrust and drag coefficients, and the horizontal distance of each rotor from the centre of gravity, respectively. Table 3 represents the value of these parameters. In comparison to previous benchmarks, the 3-DOF problem is unstable. For this purpose, two cases are supposed as follows. In the first case, we assume that all states of the quadrotor are available and we do not have any estimation. In simulations, J_{EE} is removed from Equation (13), and the augmented state space is reduced to control elements. In the second case, we assume that only roll, pitch and yaw angles are measured and other states are estimated; therefore,

$$\mathbf{y}(t) = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \mathbf{x}(t) + \boldsymbol{v}(t) \tag{37}$$

where $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t), x_4(t), x_5(t), x_6(t)]^T$ is the state vector, $\mathbf{y}(t)$ is the observation vector and $\boldsymbol{v}(t) \sim N(0, \sigma_v^2)$ is the vector of observation noise. In addition, the variance of the measurement noise is set as $\sigma_v^2 = 0.01$.

The parameters of CACC for case 1 are listed in Table 2. The initial value of states and controls are zero, and the objective is to track the desired attitude. The desired attitude is $\varphi = 40°$, $\theta = 20°$ and $\psi = -30°$. Time response of the Euler angles, the control signals and the angular rates are depicted in Figure 11.

In the second case, in addition to the control inputs, some states (angular rates) must also be estimated, where the dynamic system is unstable. In this case, the control inputs are assumed as summation of a stabilizing term $\mathbf{u}_{st}(t)$ and a tracking term $\mathbf{u}_{tr}(t)$, i.e. $\mathbf{u}(t) = \mathbf{u}_{st}(t) + \mathbf{u}_{tr}(t)$. The first term, $\mathbf{u}_{st}(t)$ stabilizes the unstable dynamic of the quadrotor for the state estimation to be accomplished properly, and the second term $\mathbf{u}_{tr}(t)$ is used to track the desired set point using CACC. The quadrotor is stabilized with non-linear state feedback,

Table 3. Constant parameters of the quadrotor (Nobahari and Sharifi, 2014).

Parameter	Unit	Value	Description
m	kg	0.8	Mass
d_{cg}	m	0.2	Horizontal distance of each rotor from the center of gravity
T_t	N	8	Total thrust
b	kgm^2	3×10^{-5}	Thrust factor
d	kgm^2	3×10^{-6}	Drag factor
I_{xx}	kgm^2	0.028	Moment of inertia around x-axis
I_{yy}	kgm^2	0.031	Moment of inertia around y-axis
I_{zz}	kgm^2	0.044	Moment of inertia around z-axis
I_{rotor}	kgm^2	8.3×10^{-5}	Moment of inertia of a rotor around the axis of rotation

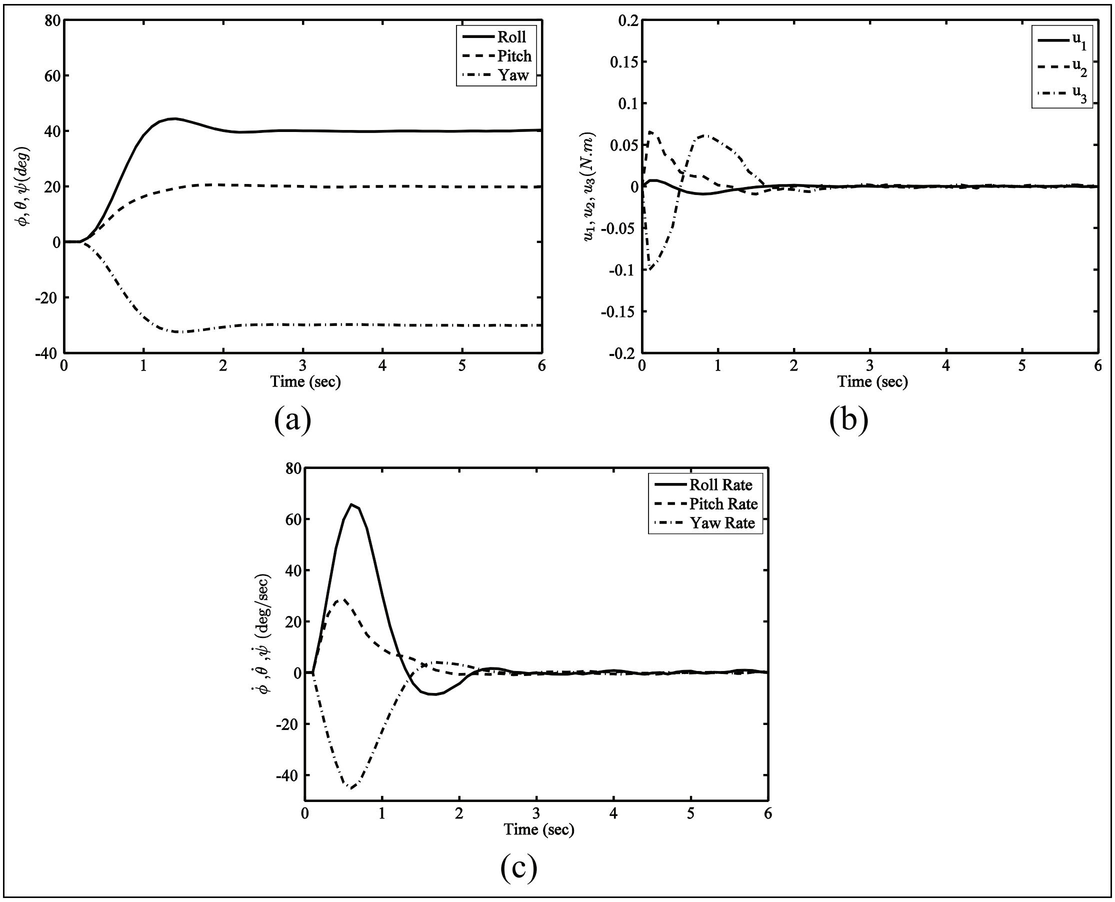


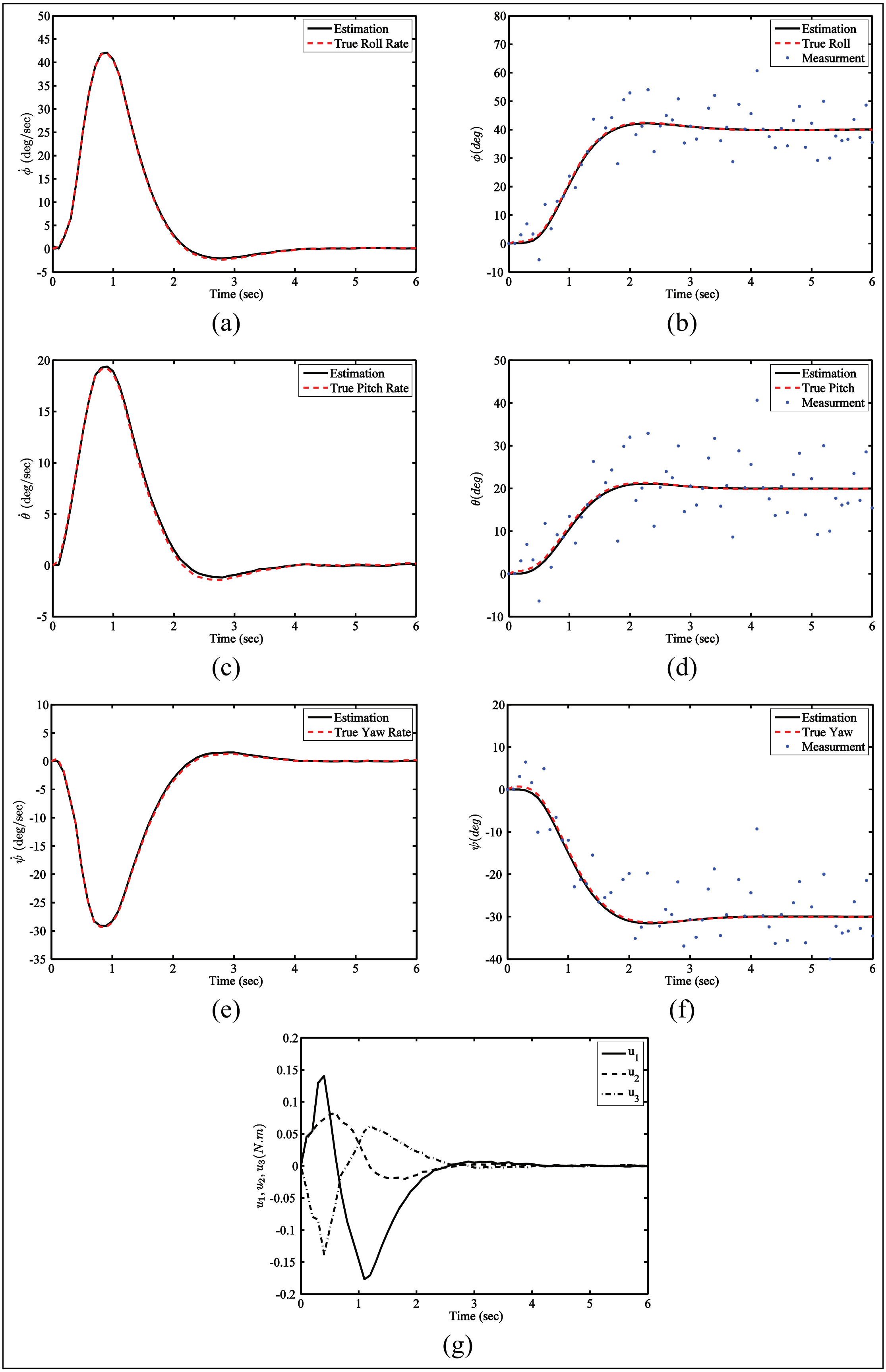
Figure 11. Performance of the controller (without estimation): (a) roll, pitch and yaw angles; (b) control signals; (c) roll, pitch and yaw rates.

$\mathbf{u}_{st}(t)$, derived based on Lyapunov theory (Slotine and Li, 1991) (see appendix A).

In this case, CACC is used for both estimation and control. Time response of the states and the control signals are depicted in Figure 12. Figure 12(a) illustrates the estimated and true roll rate. Furthermore, the estimated, true and measured roll angle are represented in Figure 12(b). The same results for pitch and yaw channels can also be observed in this figure. Moreover, Figure 12(g) represent the control signals.

The results, obtained from the above experiments, show that the proposed non-linear state estimation and control algorithm can successfully estimate the states of these systems and control the desired state with acceptable tracking error, control effort and control smoothness.

In the following, the performance of the proposed algorithm is compared with other works and the proportional–integral–derivative (PID) controller. For comparison, some common criteria, including integral of square error (ISE), integral of absolute error (IAE), integral of time multiplied



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Figure 12. Performance of continuous ant colony controller (CACC) (with estimation): (a) estimated and true roll rate; (b) estimated, true and measured roll angle; (c) pitch rate; (d) pitch angle; (e) yaw rate; (f) yaw angle; (g) control signals.

Table 4. Parameters of the proportional–integral–derivative (PID) controller.

Problem	K_P	K_I	K_D
CSTR	274.3	426.8	19.7
Cart and spring	11.5	4.9	3.7
Quadrotor	7.2	1.2	3.4

CSTR, continuous stirred tank reactor.

square error (ITSE) and integral of time multiplied absolute error (ITAE). Parameters of the PID controller, listed in Table 4, are determined using Tabu CACS (TCACS) (Karimi et al., 2010) to minimize the ISE. As presented in Table 5, results show the advantage of the proposed method in comparison with the others.

Computational cost

The authors have implemented CACC in MATLAB, and a computer with a Core 2 Duo 2.9 GHz CPU and 3 GByte RAM is used for simulation. The computing time for one step of CSTR, is approximately 0.026 s, whereas the sampling time of this problem is 3 s (0.05 min), as in Botchu and Ungarala (2007), and Stahl and Hauth (2011). In addition, the computing time for one step of the cart and spring system is approximately 0.023 s, whereas the sampling time of this problem is 0.4 s, as in Rubagotti et al. (2011) and Grancharova and Johansen (2012). The computing time for one step of the quadrotor problem is approximately 0.093 s in case 1, and 0.078 s in case 2, whereas the sampling time of this problem is 0.1 s. Moreover, implementing the proposed algorithm in C++ can decrease the reported computational times.

Conclusion

In this paper, a new heuristic sample-based controller was proposed for non-linear stochastic Gaussian/non-Gaussian systems. The proposed controller, called CACC, models the state estimation and control problems as a dynamic optimization problem and utilizes colony of virtual ants to solve this problem. An augmented model was proposed to integrate the estimation and control problems. An integrated cost function was proposed to minimize simultaneously the state estimation error, tracking error, control effort and control smoothness. CACC was tested over three non-linear problems: a non-linear CSTR, a non-linear cart and spring system, and the attitude control of a quadrotor. In the quadrotor problem, which is an unstable MIMO one, two cases were studied. In the first case, we did not have any estimation. In the second case, we assumed that only roll, pitch and yaw angles are available and the rates are estimated. In this case, a non-linear feedback was added to the control input to stabilize the system. The computational cost was evaluated for three non-linear problems, and the results show that the proposed approach is real-time capable. To investigate the effect of the parameters of the proposed controller, a sensitivity analysis was performed and the performance of the new approach was compared with other works and the PID controller using some common criteria such as ISE, IAE, ITSE and ITAE. The results demonstrate that the proposed approach can successfully estimate the states and find the optimal control of non-linear Gaussian/non-Gaussian systems, simultaneously. Obviously, there are some drawbacks. First, it is difficult to derive an analytical stability for the proposed algorithm. Furthermore, one cannot say anything about the number of ants and iterations needed for estimating the states and control signals. In some cases, one has to use so many ants and iterations to obtain good results, which can increases the computational cost.

Table 5. Comparison of the proposed controller with other approaches.

Problem	Controller	Time (min)	ISE $((mol/l)^2)$	IAE (mol/l)	ITSE $(\min .(mol/l)^2)$	ITAE $(\min .(mol/l))$
CSTR	CACC	6.5	0.0131	0.6332	0.0144	1.0897
	Botchu and Ungarala (2007)		0.0168	0.6951	0.0185	1.1510
	PID		0.0435	0.8012	0.0941	1.4023
	CACC	15	0.0140	0.6961	0.0381	2.8104
	Stahl and Hauth (2011)		0.0151	0.7257	0.0436	2.9350
	PID		0.0635	0.9360	0.1530	3.0251
	Controller	Time (s)	ISE (m^2)	IAE (m)	ITSE $(\sec .m^2)$	ITAE $(\sec .m)$
Cart and spring	CACC	6	2.8231	2.0711	1.1435	1.7240
	Rubagotti et al. (2011)		4.0071	2.9274	2.1328	2.8446
	PID		3.1694	2.2786	1.2268	1.9352
	CACC	16	1.8930	1.6970	0.7648	1.4347
	Grancharova and Johansen (2012)		2.6518	2.4287	1.4279	2.3676
	PID		2.9530	3.5057	3.2760	8.8451
	Controller	Time (s)	ISE $(\deg^2)$	IAE $(\deg)$	ITSE $(\sec .\deg^2)$	ITAE $(\sec .\deg)$
Quadrotor	CACC	10	2.6125×10^4	644.7264	1.3520×10^5	2813
	Khan and Kadri (2015) (Scenario 1)		2.8671×10^4	679.3680	1.5185×10^5	2829
	PID		3.658×10^4	691.3214	1.6380×10^5	2841

ISE, integral of square error; IAE, integral of absolute error; ITSE, integral of time multiplied square error; ITAE, integral of time multiplied absolute error; CSTR, continuous stirred tank reactor; CACC, continuous ant colony controller; PID, proportional–integral–derivative.

Acknowledgements

The authors gratefully acknowledge financial supports of the research office at Sharif University of technology.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

References

Alexis K, Nikolakopoulos G and Tzes A (2012) Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory & Applications* 6(12): 1812–1827.

Alexis K, Nikolakopoulos G and Tzes A (2011) Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice* 19(10): 1195–1207.

Bagis A (2011) Tabu search algorithm based PID controller tuning for desired system specifications. *Journal of the Franklin Institute* 348(10): 2795–2812.

Best CM, Gillespie MT, Hyatt P, et al (2015) Model predictive control for pneumatically actuated soft robots. *IEEE Transactions on Robotics and Automation* 7: 16–17.

Blackmore L, Ono M, Bektassov A, et al. (2010) A probabilistic particle-control approximation of chance-constrained stochastic predictive control. *IEEE Transactions on Robotics* 26(3): 502–517.

Botchu S and Ungarala S (2007) Nonlinear model predictive control based on sequential Monte Carlo state estimation. *IFAC Proceedings Volumes*. Available at: http://www.sciencedirect.com/science/article/pii/S1474667015317663 [Accessed 22 August 2017].

Bououden S, Chadli M and Karimi HR (2015) An ant colony optimization-based fuzzy predictive control approach for nonlinear processes. *Information Sciences* 299: 143–158.

Camacho EF and Bordons C (2007) Nonlinear model predictive control: an introductory review. In: *Assessment and Future Directions of Nonlinear Model Predictive Control*. Berlin: Springer, pp. 1–16.

Choi Y-C and Ahn H-S (2015) Nonlinear control of quadrotor for point tracking: Actual implementation and experimental tests. *IEEE/ASME Transactions on Mechatronics* 20(3): 1179–1192.

Clarke DW, Mohtadi C and Tuffs PS (1987) Generalized predictive control – Part I. The basic algorithm. *Automatica* 23(2): 137–148.

Cutler CR and Ramaker BL (1980) Dynamic matrix control a computer control algorithm. In: *Proceedings Joint Automatic Control Conference*, San Francisco, CA.

De Villiers JP, Godsill SJ and Singh SS (2011) Particle predictive control. *Journal of Statistical Planning and Inference* 141(5): 1753–1763.

Dorigo M, Maniezzo V and Colorni A (1996) Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 26(1): 29–41.

Doucet A and Johansen AM (2009) A tutorial on particle filtering and smoothing: Fifteen years later. In: Crisan D and Rozovskii (eds) *Handbook of Nonlinear Filtering*. Oxford: Oxford University Press, 12(656–704).

Eberhart RC and Kennedy J (1995) A new optimizer using particle swarm theory. In: *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39–43. Piscataway, NJ: IEEE Service Center.

Fan J and Han M (2012) Nonliear model predictive control of ball–plate system based on Gaussian particle swarm optimization. In: *IEEE Congress on Evolutionary Computation*, Brisbane, Australia, 2012, pp. 1–6.

Glover F (1989) Tabu search – part I. *ORSA Journal on Computing* 1(3): 190–206.

Goldberg DE and Holland JH (1988) Genetic algorithms and machine learning. *Machine Learning* 3(2): 95–99.

Gordon NJ, Salmond DJ and Smith AFM (1993) Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F – Radar and Signal Processing*, 107–113.

Gozde H and Taplamacioglu MC (2011) Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system. *Journal of the Franklin Institute* 348(8): 1927–1946.

Grancharova A and Johansen TA (2012) *Explicit Nonlinear Model Predictive Control: Theory and Applications*. Berlin: Springer.

Haeri M and Beik HZ (2005) Application of extended DMC for nonlinear MIMO systems. *Computers & Chemical Engineering* 29(9): 1867–1874.

Ho Y, Nguyen BP and Chui C-K (2012) Ant colony optimization for model predictive control for blood glucose regulation. In: *Proceedings of the Third Symposium on Information and Communication Technology*, Ha Long, Vietnam, pp. 214–217.

Huang J, Qian J, Liu L, et al. (2016) Echo state network based predictive control with particle swarm optimization for pneumatic muscle actuator. *Journal of the Franklin Institute* 353(12): 2761–2782.

Huyck B, De Brabanter J, De Moor B, et al. (2014) Online model predictive control of industrial processes using low level control hardware: A pilot-scale distillation column case study. *Control Engineering Practice* 28: 34–48.

Júnior GAN, Martins MAF and Kalid R (2014) A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty. *ISA Transactions* 53(2): 560–567.

Kantas N, Maciejowski JM and Lecchini-Visintini A (2009) Sequential Monte Carlo for model predictive control. In: *Nonlinear Model Predictive Control*. Berlin: Springer, pp. 263–273.

Karimi A, Nobahari H and Siarry P (2010) Continuous ant colony system and tabu search algorithms hybridized for global minimization of continuous multi-minima functions. *Computational Optimization and Applications* 45(3): 639–661.

Khan HS and Kadri MB (2015) Attitude and altitude control of quadrotor by discrete PID control and non-linear model predictive control. In: *2015 International Conference on Information and Communication Technologies (ICICT)*, Karachi, Pakistan, pp. 1–11.

Kirkpatrick S, Gelatt CD Jr, Vecchi MP, et al. (1983) Optimization by simulated annealing. *Science* 220(4598): 671–680.

Kloeden PE and Platen E (1992) *Numerical Solution of Stochastic Differential Equations*. Berlin: Springer.

Maciejowski JM (2002) *Predictive Control: With Constraints*. London: Pearson Education.

Modirrousta A and Khodabandeh M (2015) A novel nonlinear hybrid controller design for an uncertain quadrotor with disturbances. *Aerospace Science and Technology* 45: 294–308.

Najim K, Ikonen E and Del Moral P (2006) Open-loop regulation and tracking control based on a genealogical decision tree. *Neural Computing & Applications* 15(3–4): 339–349.

Nemati A and Yousefi SA (2016a) A numerical method for solving fractional optimal control problems using Ritz method. *Journal of Computational and Nonlinear Dynamics* 11(5): 51015.

Nemati A and Yousefi SA (2016b) A numerical scheme for solving two-dimensional fractional optimal control problems by the Ritz method combined with fractional operational matrix. *IMA Journal of Mathematical Control and Information* 34(4): 1079–1097.

Nobahari H and Nasrollahi S (2016) A nonlinear estimation and control algorithm based on ant colony optimization. In: *2016 IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, BC, Canada , pp. 5120–5127.

Nobahari H and Sharifi AR (2014) Continuous ant colony filter applied to online estimation and compensation of ground effect in automatic landing of quadrotor. *Engineering Applications of Artificial Intelligence* 32: 100–111.

Özkan L and Kothare M V (2006) Stability analysis of a multi-model predictive control algorithm with application to control of chemical reactors. *Journal of Process Control* 16(2): 81–90.

Panda S, Sahu BK and Mohanty PK (2012) Design and performance analysis of PID controller for an automatic voltage regulator system using simplified particle swarm optimization. *Journal of the Franklin Institute* 349(8): 2609–2625.

Pourjafari E and Mojallali H (2011) Predictive control for voltage collapse avoidance using a modified discrete multi-valued PSO algorithm. *ISA Transactions* 50(2): 195–200.

Pourtakdoust SH and Nobahari H (2004) An extension of ant colony system to continuous optimization problems. In: *Ant Colony Optimization and Swarm Intelligence*. Berlin: Springer, pp. 294–301.

Rajabi F, Rezaie B and Rahmani Z (2016) A novel nonlinear model predictive control design based on a hybrid particle swarm optimization-sequential quadratic programming algorithm: application to an evaporator system. *Transactions of the Institute of Measurement and Control* 38(1): 23–32.

Rashedi E, Nezamabadi-Pour H and Saryazdi S (2009) GSA: a gravitational search algorithm. *Information Sciences* 179(13): 2232–2248.

Richalet J, Rault A, Testud JL, et al. (1978) Model predictive heuristic control: applications to industrial processes. *Automatica* 14(5): 413–428.

Richalet J, Abu El Ata-Doss S, Arber C, et al. (1987) Predictive functional control: Application to fast and accurate robots. In: *Proceedings of the 10th IFAC Congress*, Munich, Germany, pp. 251–258.

Rossiter JA (2013) *Model-based Predictive Control: A Practical Approach*. Boca Raton, FL: CRC Press.

Rubagotti M, Raimondo DM, Ferrara A, et al. (2011) Robust model predictive control with integral sliding mode in continuous-time sampled-data nonlinear systems. *IEEE Transactions on Automatic Control* 56(3): 556–570.

Rubinstein RY and Kroese DP (2016) *Simulation and the Monte Carlo Method*. New York: John Wiley & Sons.

Salmond D and Gordon N (2001) Particles and mixtures for tracking and guidance. In: *Sequential Monte Carlo Methods in Practice*. Berlin: Springer, pp. 517–532.

Sarailoo M, Rahmani Z and Rezaie B (2015) A novel model predictive control scheme based on bees algorithm in a class of nonlinear systems: application to a three tank system. *Neurocomputing* 152: 294–304.

Sileshi BG, Ferrer C and Oliver J (2013) Particle filters and resampling techniques: importance in computational complexity analysis. In: *2013 Conference on Design and Architectures for Signal and Image Processing (DASIP)*, Cagliari, Italy, pp. 319–325.

Slotine J-JE and Li W (1991) *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall.

Smoczek J and Szpytko J (2014) Evolutionary algorithm-based design of a fuzzy TBF predictive model and TSK fuzzy anti-sway crane control system. *Engineering Applications of Artificial Intelligence* 28: 190–200.

Van Soest WR, Chu QP and Mulder JA (2006) Combined feedback linearization and constrained model predictive control for entry flight. *Journal of Guidance, Control, and Dynamics* 29(2): 427–434.

Stahl D and Hauth J (2011) PF-MPC: particle filter-model predictive control. *Systems & Control Letters* 60(8): 632–643.

Stutzle T and Dorigo M (2002) A short convergence proof for a class of ant colony optimization algorithms. *IEEE Transactions on Evolutionary Computation* 6(4): 358–365.

Varziri MS, Poyton AA, McAuley K, et al. (2008) Selecting optimal weighting factors in iPDA for parameter estimation in continuous-time dynamic models. *Computers & Chemical Engineering* 32(12): 3011–3022.

Yin S, Gao H and Qiu J (2017a) Descriptor reduced-order sliding mode observers design for switched systems with sensor and actuator faults. *Automatica* 76: 282–292.

Yin S, Yang H and Kaynak O (2017b) Sliding mode observer-based FTC for Markovian jump systems with actuator and sensor faults. *IEEE Transactions on Automatic Control* 62(7): 3551–3558.

Appendix A

Consider the following positive definite Lyapunov function candidate:

$$V(t) = \frac{1}{2}\left(x_1^2(t) + x_2^2(t) + x_3^2(t) + x_4^2(t) + x_5^2(t) + x_6^2(t)\right) \quad \text{(A.1)}$$

The derivative of Lyapunov function candidate is rewritten as:

$$\begin{aligned}\dot{V}(t) = {} & x_1(t)\dot{x}_1(t) + x_2(t)\dot{x}_2(t) + x_3(t)\dot{x}_3(t) + x_4(t)\dot{x}_4(t) \\ & + x_5(t)\dot{x}_5(t) + x_6(t)\dot{x}_6(t)\end{aligned} \quad \text{(A.2)}$$

If the control inputs are chosen as:

$$u_{1\mathrm{st}}(t) = \\ I_{\mathrm{xx}}\left(-\frac{(I_{\mathrm{yy}} - I_{\mathrm{zz}})}{I_{\mathrm{xx}}}x_4(t)x_6(t) - \frac{I_{\mathrm{rotor}}}{I_{\mathrm{xx}}}\Omega_{\mathrm{r}}x_4(t) - x_1(t) - x_2(t) - x_2^3(t)\right) \quad \text{(A.3)}$$

$$u_{2\mathrm{st}}(t) = \\ I_{\mathrm{yy}}\left(-\frac{(I_{\mathrm{zz}} - I_{\mathrm{xx}})}{I_{\mathrm{yy}}}x_2(t)x_6(t) + \frac{I_{\mathrm{rotor}}}{I_{\mathrm{yy}}}\Omega_{\mathrm{r}}x_4(t) - x_3(t) - x_4(t) - x_4^3(t)\right) \quad \text{(A.4)}$$

$$u_{3\mathrm{st}}(t) = I_{\mathrm{zz}}\left(-\frac{(I_{\mathrm{xx}} - I_{\mathrm{yy}})}{I_{\mathrm{zz}}}x_2(t)x_4(t) - x_5(t) - x_6(t) - x_6^3(t)\right) \quad \text{(A.5)}$$

The derivative of Lyapunov function, $\dot{V}(t) = -x_2^2(t) - x_2^4(t) - x_4^2(t) - x_4^4(t) - x_6^2(t) - x_6^4(t)$, is semi negative definite. In simulation, states, $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t), x_4(t), x_5(t), x_6(t)]^{\mathrm{T}}$, are replaced with estimated states, $\hat{\mathbf{x}}(t) = [\hat{x}_1(t), \hat{x}_2(t), \hat{x}_3(t), \hat{x}_4(t), \hat{x}_5(t), \hat{x}_6(t)]^{\mathrm{T}}$.