Robust analysis for data-driven model predictive control

Hong Jianwang\(^a\), Ricardo A. Ramirez-Mendoza\(^b\) and Tang Xiaojun\(^a\)

\(^a\)School of Electronic Engineering and Automation, Jiangxi University of Science and Technology, Ganzhou, People’s Republic of China; \(^b\)School of Engineering and Sciences, Tecnologico de Monterrey, Monterrey, Mexico

ABSTRACT

Here the idea of data driven is introduced in model predictive control to establish our proposed data-driven model predictive control. Considering one first-order discrete time nonlinear dynamical system, the main essence of data driven means the actual output value in cost function for model predictive control is identified through input–output observed data in case of unknown but bounded noise and martingale difference sequence. After substituting the identified actual output in cost function, the total cost function in model predictive control is reformulated as its standard form, i.e. one quadratic program problem with input and output constraints. Then semidefinite relaxation scheme is used to derive a lower bound for its optimal value, and the robust counterpart of an uncertain quadratic program is reduced to one conic quadratic problem. The above semidefinite relaxation scheme and conic quadratic problem correspond to the similar robust analysis based on convex optimization theory. Finally, one simulation example is used to prove the efficiency of our proposed theory.

1. Introduction

Model predictive control (MPC) is a powerful methodology that has been widely considered and used in a variety of industrial applications, such as chemical process, water networks as well as building energy managements. More specifically, model predictive control is one special form of suboptimal control problem, whose control objective is to keep the state of a system near some desired points. MPC combines elements of several ideas that we have put forth, for example, certainty equivalent control, multistage lookahead and rollout algorithms. MPC tends some applied properties for classical linear quadratic control, i.e. there are two main reasons for replacing that classical linear quadratic control by MPC. (1) The considered system may be nonlinear, and a model that is linearized around the desired point may be inappropriate. (2) There may be control and state constraints, which are not handled adequately through a quadratic penalty on state and control. The solution obtained from a linear quadratic model is not suitable for this, because the quadratic penalty on state and control tends to blur the boundaries of the constraints. Generally, MPC converts one optimal control problem into one numerical optimization problem with equality or inequality constraints, which correspond to the control and state constraints. Moreover, when the considered system is either deterministic, or else it is stochastic, it is replaced with a deterministic version by using typical values in place of all uncertain quantities, such as the certainty control approach in implementing MPC. Roughly speaking, at each stage, an optimal control problem is solved over a fixed length horizon, starting from the current state. The first component of the corresponding optimal policy is then used as the control of the current stage, while the remaining components are discarded. The optimization process is then repeated at the next stage, once the next state is revealed or the optimization algorithm is terminated iteratively.

From above detailed description on MPC, MPC corresponds to one numerical optimization problem, whose cost function or loss function is one error value between the actual output and its desired output reference. In reality, the desired output is given, but the actual output is unknown in priori, so first we need to model the considered system and collect its actual output through persistly exciting the system by one appropriate input signal. It means the considered system is identified and used to calculate the actual output. There are two modelling approaches, used to identify the considered system, i.e. the first principle and system identification. The first principle needs lots of priori information about the considered system, such as Newton law, mathematical or physical laws, and so on. Then main essence of system
identification is to excite the considered system, then use these collected input–output data to identify or estimate the unknown parameters, as the parameters are estimated online and used to describe the considered system, whatever in open loop or closed loop condition. The advantage of the second system identification approach is that no any priori information are needed, but only input–output data. In this big data period, this requirement for input–output data is tolerable. Roughly speaking, when to obtain the actual output in cost function for MPC, the input–output data, corresponding to the considered system in open loop or closed loop, are collected to identify the system through some statistical methods, for example, least squares method, maximum likelihood method, etc. Then the identified system is applied to express or describe the actual output, so the actual output depends on the accuracy of the identified system. In practice, system identification is a well-developed technique for estimating system parameters from operational data, typically taken during dedicated system testing or excitation, so system identification is also named as data driven.

Due to the application of system identification into MPC or other control strategy, such as adaptive control, internal model control and robust control, a new concept identification for control was proposed in 2010s. Here we give a concise introduction or contribution on identification for control. In case of the unknown but bounded noise, one bounded error identification is proposed to identify the unknown systems with time varying parameters. Then one feasible parameter set is constructed to include the unknown parameter with a given probability level. In Alamo et al. (2009), the feasible parameter set is replaced by one confidence interval, as this confidence interval can accurately describe the actual probability that the future predictor will fall into the constructed confidence interval. The problem about how to construct this confidence interval is solved by a linear approximation/programming approach, which can identify the unknown parameter only for linear regression model. According to the obtained feasible parameter set or confidence interval, the midpoint or centre can be deemed as the final parameter estimation, further a unified framework for solving the centre of the confidence interval is modified to satisfy the robustness. This robustness corresponds to other external noises, such as outlier, unmeasured disturbance (Bertsimas & Goyal, 2012). The above-mentioned identification strategy, used to construct one set or interval for unknown parameter, is called as set membership identification, dealing with the unknown but bounded noise. There are two types of descriptions on external noise, one is probabilistic description, the other is deterministic description, corresponding to the unknown but bounded noise here (Blackmore et al., 2011). For the probabilistic description on external noise, the noise is always assumed to be one white noise, and its probabilistic density function (PDF) is known in advance. On the contrary for deterministic description on external noise, the only information about noise is bound, so this deterministic description can relax the strict assumption on probabilistic description. In reality or practice, bounded noise is more common than white noise. Within the deterministic description on external noise, set membership identification is adjusted to design controllers with two degrees of freedom (Calafiore, 2010), it corresponds to data-driven control or set membership control. Set membership control is applied to design feedback control in a closed loop system with nonlinear system in Campi and Garatti (2016), where the considered system is identified by set membership identification, and the obtained system parameter will be benefit for the prediction output. After substituting the obtained system parameter into the prediction output to construct one cost function, reference (Campi et al., 2009) takes the derivative of the above cost function with respect to control input to achieve one optimal input. Set membership identification can be not only applied in MC, but also in stochastic adaptive control (Callaway & Hiskens, 2011), where a learning theory kernel is introduced to achieve the approximation for nonlinear function or system. Based on the bounded noise, many parameters are also included in known intervals in prior, then robust optimal control with adjustable uncertainty sets is studied in Zhang et al. (2017), where robust optimization is introduced to consider uncertain noise and uncertain parameter simultaneously. To solve the expectation operation with dependence on the uncertainty, sample size of random convex programs is considered to replace the expectation by finite sum (Zhang et al., 2015). Generally, many practical problems in systems and control, such as controller synthesis and state estimation, are often formulated as optimization problems (Garatti & Campi, 2013). In many cases, the cost function incorporates variables that are used to model uncertainty, in addition to optimization variables, and reference (Farina et al., 2016) employs uncertainty described as probabilistic variables. Generally, the above-mentioned references are divided into two types, i.e. system identification and model predictive control, and these two types are separated to each other in today’s research. Due to their separations, the combinations of them are considered now, i.e. identification for control. The mission of this paper is to achieve this goal, i.e. combining system identification and model predictive control. More specifically, system identification is proposed to identify the nonlinear function estimation, then this obtained nonlinear function
estimation is substituted into the cost function for model predictive control. Finally, robust analysis is further studied from the point of bounded noise and martingale difference noise.

Here in this paper, we apply our previous work on robust control or robust optimization into this model predictive control, and in real-world applications, the data are not always known exactly, i.e. we know about the data is that they belong to a given set. From the theoretical perspective, an uncertain quadratic programming for model predictive control is formulated through our own derivations, then its robust counterpart of this uncertain quadratic programming corresponds to our studied robust analysis for model predictive control. More specifically, consider one first-order discrete time nonlinear dynamic control system, its output must be needed in the cost function for MPC, so first we need to construct the actual output for the considered nonlinear dynamical system. But here it is very different with the references, as the nonlinear dynamical system is considered. To implement the proposed MPC well that actual output is identified for the nonlinear dynamical system., Moreover, our own derivations belong to the idea of system identification, i.e. the input–output data are proposed to construct the actual output directly, not to identify the nonlinear system parameter. Roughly speaking, data are used to describe the actual output directly, thus avoiding to estimate the nonlinear system parameter. The process of using the input–output data to denote the actual output directly coincides to the essence of data driven. After substituting the obtained actual output in the cost function for MPC, one numerical optimization problem with input and output constraints is needed to solve. After simple but tedious calculation, this constrain optimization problem can be formulated as one standard quadratic program problem. Due to numerous problems of planning, scheduling, routing, etc. can be posed as combinatorial optimization problems, i.e. optimization programs with discrete design variables. Then a combinatorial problem can be posed as minimizing a quadratic objective under quadratic inequality constraints. Lagrange relaxation or dual theory is proposed to analyse this quadratic program problem with inequality constraints, further this problem is reduced to one conic quadratic program problem in case of uncertain data, i.e. the data are typically known in the space of data- and uncertainty set, which for sure contains the actual data. This case exists in reality, where in spite of this data uncertainty, our decision must satisfy the actual constraints, whether we know them or not. Throughout this paper, the idea about applying system identification theory to identify or construct the actual output for one discrete time nonlinear dynamic control system corresponds to the data driven for model predictive control. Similarly consider the obtained quadratic program problem with inequality constraints, semidefinite relaxation scheme is proposed to analyse a lower bound for the optimal value, and the robust counterpart of an uncertain quadratic program is studied to be one conic quadratic problem. This robust counterpart coincides with the robust analysis for the above obtained data-driven model predictive control.

More specifically, in this paper, we mainly concern the following two contributions.

(1) In case of the unknown but bounded noise and martingale difference noise, the nonlinear function estimation, corresponding to the unknown nonlinear function, is constructed based on the input–output data.

(2) After substituting this obtained nonlinear function estimation into the cost function in model predictive control, one quadratic program problem with input and output constraints is reformulated.

(3) Consider the robust counterpart of an uncertain quadratic program, the semidefinite relaxation scheme is applied to get a lower bound.

This paper is organized as follows. In Section 2, one first-order discrete time nonlinear dynamical control system is considered, and some preliminaries are formulated about the noise for the nonlinear function. In Section 3, the estimate of the unknown nonlinear function is studied. Two estimations for the unknown nonlinear function are derived based on the detailed noise, i.e. unknown but bounded noise and martingale difference noise. This process about constructing the unknown nonlinear function corresponds to the nonlinear function estimation based on input–output data. It means we apply the collected input–output data to identify the unknown nonlinear function. In Section 4, the obtained estimation for the unknown nonlinear function is substituted into the cost function in model predictive control. This cost function is formulated or rewritten as one quadratic program problem in case of input and output constraints. Throughout our own derivations, it is reduced to one quadratic program problem with inequality constraints. Then semidefinite relaxation scheme is used to show a lower bound for its optimal value in Section 5, where the robust counterpart of an uncertain quadratic program is changed to be one conic quadratic problem. The whole analysis process is our named as our robust analysis. In Section 6, one simulation example illustrates the effectiveness of the proposed theories. Section 7 ends the paper with final conclusion and points out the next subject of ongoing research. Here in this paper, all the mathematical derivations are obtained by our own contributions.
2. Nonlinear dynamic system

To give an explicit expression about the actual output in one cost function for our considered model predictive control, we need to describe this actual output for one considered system, whatever linear or nonlinear dynamic system.

As linear system is widely studied in lots of references, so here we consider a more general case, i.e. consider the following first-order discrete time nonlinear dynamical system:

\[ y(t + 1) = f(y(t)) + u(t) + w(t + 1), \quad t \geq 0 \]  

(1)

where in Equation (1), \( y(t) \) and \( u(t) \) are the system output and input respectively at time instant \( t \). Nonlinear function \( f() \) is completely unknown. Equation (1) shows the closed relation between two adjacent time instants \( (t + 1) \) and \( t \). To be convenient for the later mathematical derivations, here the right side of Equation (1) considers the external noise \( w(t + 1) \). The goal of next section is to estimate or identify this nonlinear function from input–output data sequence \( \{u(t), y(t)\} \), external noise \( w(t) \) is one unknown but bounded noise, which extends the special case of white noise, and its upper bound \( w > 0 \) satisfies.

\[ \|w(t)\| \leq w, \quad \forall t \geq 0 \]  

(2)

To apply MPC to design the predictive controller \( u(t) \), the estimation of the nonlinear function \( f(y(t)) \) is expected to track one desired output reference \( y_{\text{des}}(t) \). To measure its discrepancy, error value \( (f(y(t)) - y_{\text{des}}(t)) \) is needed to expand at time instant \( t \). Due to nonlinear function \( f(y(t)) \) is unknown, so the urgent mission is to estimate this nonlinear function and use its estimation \( \hat{y}(t) \) in error value, i.e. \( (\hat{y}(t) - y_{\text{des}}(t)) \).

In case of unknown but bounded noise \( w(t) \), the nearest neighbour estimation for nonlinear function \( f() \) is described to achieve the tracking.

Set

\[
\begin{align*}
\hat{b}(t) &= \max_{0 \leq i \leq t} y(i) \\
\bar{b}(t) &= \min_{0 \leq i \leq t} y(i)
\end{align*}
\]

(3)

and

\[ i_t = \arg \min_{0 \leq i \leq t-1} \|y(t) - y(i)\| \]  

(4)

Then we set

\[ \|y(t) - y(i_t)\| = \min_{0 \leq i \leq t-1} \|y(t) - y(i)\|, \quad t \geq 1 \]  

(5)

So at each time instant \( t \geq 1 \), the nearest neighbour estimation for nonlinear function \( f(y(t)) \) is given as

\[ \hat{f}(y(t)) = y(i_t + 1) - u(i_t) \]  

(6)

Equation (6) can be rewritten as

\[ \hat{f}(y(t)) = f(y(i_t + 1)) + w(t + 1) \]  

(7)

To make the actual output \( y(t) \) track the desired output reference \( y_{\text{des}}(t) \), we define

\[
\begin{align*}
u'(t) &= -\hat{f}(y(t)) + \frac{1}{2}(\bar{b}(t) + \hat{b}(t)) \\
u''(t) &= -\hat{f}(y(t)) + y_{\text{des}}(t), \quad t \geq 1
\end{align*}
\]

(8)

Based on the definition (8), then the controller is defined as follows:

\[ u(t) = \begin{cases} 
u'(t) & \text{if } \|y(t) - y(i)\| > \epsilon \\ 
u''(t) & \text{if } \|y(t) - y(i)\| \leq \epsilon \\ \end{cases} \]  

(9)

where \( \epsilon > 0 \) is one arbitrary small positive value.

We remark that the estimators (5) and (6) may be referred to as the nearest neighbour estimator for \( f() \), as can be seen intuitively from Equation (7). It is a natural one when we only know the generalized Lipschitz continuity of \( f() \) and the boundedness of the noise \( \{w(t)\} \). Using the above controller \( u(t) \) (9), the tracking mission will reach, i.e. \( \|y(t) - y(i)\| \to 0 \) and observing above description, the estimation for nonlinear function is important in determining the controller. To return back to our considered MPC, this estimation for nonlinear function is also needed, so in Section 3, we derive other estimation for nonlinear function based on our own derivation. Equation (8) is only used in tracking control not our mentioned model predictive control. Here we only give one example about constructing one nearest neighbour estimation for nonlinear function for the tracking mission. Then through using this idea same, one new nonlinear function estimation is studied for our proposed model predictive control.

3. Nonlinear function estimation

The nearest neighbour estimation (6) is efficient in case of bounded noise (2). To this end, as a consequence we assume that noise \( w(t) \) is a martingale difference sequence, i.e.

\[ E[w(t + 1)|B_t] = 0, \quad t \geq 0 \]  

(10)

where \( E \) means the expectation, and \( B_t = \sigma\{w(0), \ldots w(t)\} \) is \( \sigma \)-algebra, generated by sequence \( \{w(0), \ldots w(t)\} \).
Making use of the property of martingale difference sequence, set

\[ \delta_j = [\epsilon, (j + 1)\epsilon], \quad j \in Z = \{ \text{all integers} \} \tag{11} \]

where \( Z \) is the set of all integers, \( \epsilon > 0 \) is one arbitrary small positive value, and it holds that.

\[ \bigcup_{j \in Z} = (-\infty, +\infty) \tag{12} \]

and

\[ \delta_i \bigcap \delta_j = \phi, \quad \text{for } i \neq j \tag{13} \]

We define the following interval value function \( \Delta(\cdot) \) as, for each \( y \)

\[ \Delta(y) = \delta_{j-1} \cup \delta_j \cup \delta_{j+1}, \quad \text{if } y \in \delta_j, j \in Z \tag{14} \]

It means \( \Delta(y) \) covers the \( \epsilon \) neighbourhood of \( y \). For every \( t \geq 1, \|y(t) - y(i)\| \leq \epsilon \), then it holds that.

\[ \sum_{i=0}^{t-1} l_{\Delta(y(i))}(y(i)) > 0 \tag{15} \]

where \( l_{\Delta(y(i))} \) is the indicator function. Therefore, if \( \|y(t) - y(i)\| \leq \epsilon \), the estimation of that nonlinear function \( f(\cdot) \) is defined as

\[ \hat{y}(t) = \hat{r}(y(t)) = \frac{\sum_{i=0}^{t-1} (y(i + 1) - u(i)) l_{\Delta(y(i))}(y(i))}{\sum_{i=0}^{t-1} l_{\Delta(y(i))}(y(i))} \tag{16} \]

Equation (16) shows the relation to construct that nonlinear function estimation \( \hat{r}(y(t)) \) through using the input–output data \( \{u(i), y(i)\}_{i=0}^{t-1} \). To give a more detailed description on the above estimation, we list some special estimations for \( t = 1, 2, 3 \ldots N \), where \( N \) is a finite time horizon, i.e.

\[ \hat{y}(1) = \frac{(y(1) - u(0)) l_{\Delta(y(1))}(y(0))}{l_{\Delta(y(1))}(y(0))} = y(1) - u(0) \]

\[ \hat{y}(2) = \frac{(y(1) - u(0)) l_{\Delta(y(2))}(y(0))}{l_{\Delta(y(2))}(y(0)) + l_{\Delta(y(2))}(y(1))} \]

\[ \frac{(y(2) - u(1)) l_{\Delta(y(2))}(y(1))}{l_{\Delta(y(2))}(y(0)) + l_{\Delta(y(2))}(y(1))} + \frac{l_{\Delta(y(2))}(y(0))}{l_{\Delta(y(2))}(y(0)) + l_{\Delta(y(2))}(y(1))} \]

\[ \hat{y}(3) = \frac{(y(1) - u(0)) l_{\Delta(y(3))}(y(0))}{l_{\Delta(y(3))}(y(0)) + l_{\Delta(y(3))}(y(1)) + l_{\Delta(y(3))}(y(2))} \]

\[ \frac{(y(2) - u(1)) l_{\Delta(y(3))}(y(1))}{l_{\Delta(y(3))}(y(0)) + l_{\Delta(y(3))}(y(1)) + l_{\Delta(y(3))}(y(2))} + \frac{l_{\Delta(y(3))}(y(0))}{l_{\Delta(y(3))}(y(0)) + l_{\Delta(y(3))}(y(1)) + l_{\Delta(y(3))}(y(2))} \]

\[ \frac{(y(3) - u(2)) l_{\Delta(y(3))}(y(2))}{l_{\Delta(y(3))}(y(0)) + l_{\Delta(y(3))}(y(1)) + l_{\Delta(y(3))}(y(2))} + \frac{l_{\Delta(y(3))}(y(0))}{l_{\Delta(y(3))}(y(0)) + l_{\Delta(y(3))}(y(1)) + l_{\Delta(y(3))}(y(2))} \]

\[ : \]

\[ \hat{y}(N) = \frac{(y(1) - u(0)) l_{\Delta(y(N))}(y(0))}{l_{\Delta(y(N))}(y(0)) + \cdots + l_{\Delta(y(N))}(y(N - 1))} \]

\[ + \cdots \]

\[ + \frac{(y(N) - u(N - 1)) l_{\Delta(y(N))}(y(N - 1))}{l_{\Delta(y(N))}(y(0)) + \cdots + l_{\Delta(y(N))}(y(N - 1))} \]

Based on our derived estimation of that nonlinear function, then \( \hat{y}(t) \) can be regarded as the actual output at time instant \( t \). Furthermore \( \{u(0), y(0)\} \) is initial data sequence.

### 4. Model predictive control

The goal of MPC is to make the considered system to track that desired output reference \( y_{\text{des}}(t) \) and reject noise from \( t = 0 \) up to a finite time horizon \( N \). MPC turns one optimal control problem into one numerical optimization problem, whose cost function is always set as that.

\[ \min_{u(0) \ldots u(N)} \sum_{t=0}^{N} [\hat{y}(t) - y_{\text{des}}(t)]^T Q[\hat{y}(t) - y_{\text{des}}(t)] \]

\[ + u(t)^T S u(t) \tag{17} \]

where \( \hat{y}(t) \) is the actual output, coming from Equation (16), \( y_{\text{des}}(t) \) is the desired output reference, \( Q \) and \( S \) are two positive definite weighting matrices.

For convenience, that cost function (17) is simplified to satisfy the standard form for the next analysis process. The main step in MPC is to obtain those optimal control input or optimal controller \( \{u(0), u(1), \ldots, u(N)\} \), while minimizing the cost function.

Expanding the cost function (17) to that

\[ \sum_{t=0}^{N} \hat{y}(t)^T Q \hat{y}(t) - 2y_{\text{des}}(t) Q \hat{y}(t) + y_{\text{des}}(t)^T Q y_{\text{des}}(t) \]

\[ + u(t)^T S u(t) \tag{18} \]

Neglecting the third term \( y_{\text{des}}(t)^T Q y_{\text{des}}(t) \), as it is independent of the control input \( \{u(0), u(1), \ldots, u(N)\} \), then the simplified cost function is that

\[ \sum_{t=0}^{N} \hat{y}(t)^T Q \hat{y}(t) - 2y_{\text{des}}(t) Q \hat{y}(t) + u(t)^T S u(t) \tag{19} \]

where the actual output \( \hat{y}(t) \) is given as one explicit function about the control input, i.e.

\[ \hat{y}(t) = \frac{\sum_{i=0}^{t-1} (y(i + 1) - u(i)) l_{\Delta(y(i))}(y(i))}{\sum_{i=0}^{t-1} l_{\Delta(y(i))}(y(i))} \]

A common way to specify MPC is through the use of a quadratic cost function with linear constraints, which
To simplify that cost function in Equation (21), we have

\[ u_{\text{min}} \leq u(t) \leq u_{\text{max}} \quad \hat{y}_{\text{min}} \leq \hat{y}(t) \leq \hat{y}_{\text{max}} \quad \text{(20)} \]

where \( u_{\text{min}} \) and \( u_{\text{max}} \) are lower bound and upper bound for control input \( u(t) \), \( t = 0, 1, \ldots, N \) respectively, and the definition is similar to \( \hat{y}_{\text{min}} \) and \( \hat{y}_{\text{max}} \).

Combining Equations (19) and (20), our considered model predictive control is the following optimization problem with inequality constraints.

\[
\begin{align*}
\min_{u(0) \ldots u(N)} & \quad \sum_{t=0}^{N} [\hat{y}(t) - y_{\text{des}}(t)]^{T} Q[\hat{y}(t) - y_{\text{des}}(t)] \\
& \quad + u(t)^{T} Su(t) \\
= & \sum_{t=0}^{N} \hat{y}(t) y_{\text{des}}(t) + 2y_{\text{des}}(t) \hat{y}(t) \\
& \quad + u(t)^{T} Su(t) \\
\text{subject to} & \quad u_{\text{min}} \leq u(t) \leq u_{\text{max}} \\
& \quad \hat{y}_{\text{min}} \leq \hat{y}(t) \leq \hat{y}_{\text{max}}, \quad t = 0, 1, 2 \ldots N \quad \text{(21)}
\end{align*}
\]

To simplify that cost function in Equation (21), we have

\[
\begin{align*}
\sum_{t=0}^{N} u(t)^{T} Su(t) \\
= & \begin{bmatrix} u(0) & u(1) & \cdots & u(N) \end{bmatrix} \\
\times & \begin{bmatrix} S & 0 & \cdots & 0 \\ 0 & S & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & S \end{bmatrix} \\
= & \begin{bmatrix} u(0) \\ u(1) \\ \vdots \\ u(N) \end{bmatrix} \\
\times & \begin{bmatrix} y_{\text{des}}(0) \\ y_{\text{des}}(1) \\ \vdots \\ y_{\text{des}}(N) \end{bmatrix} \\
= & b^{T} x \quad \text{(22)}
\end{align*}
\]

where \( I_{N} \) is the identity matrix with dimension \( N \).

\[
\begin{align*}
\sum_{t=0}^{N} \hat{y}(t) y_{\text{des}}(t) & = \begin{bmatrix} \hat{y}(0) & \hat{y}(1) & \cdots & \hat{y}(N) \end{bmatrix} \\
& \begin{bmatrix} Q & 0 & \cdots & 0 \\ 0 & Q & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & Q \end{bmatrix} \\
& = \begin{bmatrix} \hat{y}(0) \\ \hat{y}(1) \\ \vdots \\ \hat{y}(N) \end{bmatrix} \\
& = c^{T} y \quad \text{(23)}
\end{align*}
\]

where vector \( u \) and \( y \) are defined as follows:

\[
\begin{align*}
& u = [u(0) \ u(1) \ \cdots \ u(N)]^{T} \\
& = [\hat{y}(0) \ \hat{y}(1) \ \cdots \ \hat{y}(N)]
\end{align*}
\]

To simplify notation, set \( x = [y, u] \), then

\[
\sum_{t=0}^{N} \hat{y}(t) y_{\text{des}}(t) + \sum_{t=0}^{N} u(t)^{T} Su(t) = y^{T} Q_{Ny} + u^{T} S_{Nu}
\]

\[
= \begin{bmatrix} y & u \end{bmatrix} \begin{bmatrix} Q_{N} & 0 \\ 0 & S_{N} \end{bmatrix} \begin{bmatrix} y \\ u \end{bmatrix} = x^{T} Ax \quad \text{(24)}
\]

Similarly we get

\[
\sum_{t=0}^{N} \hat{y}(t) y_{\text{des}}(t) = \begin{bmatrix} \hat{y}(0) & \hat{y}(1) & \cdots & \hat{y}(N) \end{bmatrix} \begin{bmatrix} y_{\text{des}}(0) \\ y_{\text{des}}(1) \\ \vdots \\ y_{\text{des}}(N) \end{bmatrix} = Q \begin{bmatrix} y_{\text{des}}(0) \\ y_{\text{des}}(1) \\ \vdots \\ y_{\text{des}}(N) \end{bmatrix} 0 0
\]

\[
\begin{bmatrix} \hat{y}(0) \\ \hat{y}(1) \\ \vdots \\ \hat{y}(N) \\ u(0) \\ u(1) \\ \vdots \\ u(N) \end{bmatrix} = -b^{T} x \quad \text{(25)}
\]

where

\[
-b^{T} = Q \begin{bmatrix} y_{\text{des}}(0) & y_{\text{des}}(1) & \cdots & y_{\text{des}}(N) & 0 & 0 & \cdots & 0 \end{bmatrix}
\]

Through these two equations (24) and (25), that cost function in (21) can be rewritten as

\[
\sum_{t=0}^{N} \hat{y}(t) y_{\text{des}}(t) + 2y_{\text{des}}(t) \hat{y}(t) + u(t)^{T} Su(t)
\]
\[ = x^T Ax + 2b^T x \]  
(26)

These inequality constraints on input and output can be also reformulated as follows:

\[
\begin{bmatrix}
1 & 0 & \cdots & 0 \\
0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
u(0) \\
u(1) \\
\vdots \\
u(N)
\end{bmatrix} \leq
\begin{bmatrix}
u_{\max} \\
u_{\max} \\
\vdots \\
u_{\max}
\end{bmatrix}
\]

Similar to

\[
-\begin{bmatrix}
u_{\min} & \cdots & \cdots & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & \cdots & \cdots & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
u_{\min} \\
u_{\min} \\
\vdots \\
u_{\min}
\end{bmatrix}
\]

Combining above two inequalities to obtain

\[
\begin{bmatrix}
1 & \cdots & \cdots & \cdots & \cdots \\
0 & \cdots & \cdots & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
u \\
y
\end{bmatrix} \leq
\begin{bmatrix}
u_{\max} \\
u_{\max}
\end{bmatrix}
\]

(29)

Define \( A_i = \begin{bmatrix} 0 & I \\ I & 0 \end{bmatrix} \), as the \( i \)th column, and \( c_i = \begin{bmatrix} u_{\max} \\ -u_{\min} \end{bmatrix} \), then Equation (29) is reduced to

\[
A_i x \leq c_i, \quad i = 1, 2 \ldots N
\]

(30)

Applying the simplified cost function (26) and the simplified inequality constraint (30), the following quadratic program problem is obtained:

\[
\min_x x^T Ax + 2b^T x \\
\text{subject to} \quad A_i x \leq c_i, \quad i = 1, 2 \ldots N
\]

(31)

When to solve this above quadratic program problem (31), many existed optimization methods can be applied directly, such as Newton method, Gauss–Seidel algorithm, ADMN and our studied dynamic program. But in the next section, we do not consider the problem of how to solve this quadratic program problem with inequality constraint, but give a preliminary robust analysis.

5. Robust analysis

Observing that quadratic program problem with inequality constraint \( A_i x \leq c_i, \quad i = 1, 2 \ldots N \) again, there are numerous universal forms of combinatorial problems. It means a combinatorial problem can be posed as minimizing a quadratic objective under inequality constraints.

5.1. Semidefinite relaxation scheme

To bound from below the optimal value in Equation (31), we may use the Lagrange relaxation scheme and choose weights \( \lambda_i \geq 0, \quad i = 1, 2 \ldots N \) and add the inequality constraints with these weights to the cost function, thus coming to the following Lagrange function:

\[
L_\lambda(x) = x^T Ax + 2b^T x + \sum_{i=1}^{N} \lambda_i (A_i x - c_i)
\]

(32)

where

\[
b^T(\lambda) = b + \sum_{i=1}^{N} \frac{1}{2} \lambda_i A_i c_i(\lambda) = \sum_{i=1}^{N} \lambda_i(-c_i)
\]

By construction, the Lagrange function \( L_\lambda(x) \) is \( \leq \) the actual objective \( x^T Ax + 2b^T x \). Consequently, the unconstrained infimum of this Lagrange function

\[
\inf_x L_\lambda(x)
\]

is a lower bound for the optimal value in \( x^T Ax + 2b^T x \).

From the dual theory, we assume that \( \lambda \in R^+ \) and \( \xi \in R \) are such that.

\[
L_\lambda(x) - \xi \geq 0, \quad \forall x
\]

(33)

Recalling the structure of Lagrange function \( L_\lambda(x) \), it means that the inhomogeneous quadratic form

\[
f_\xi(x) = x^T Ax + 2b^T(\lambda)x + c(\lambda) - \xi
\]

(34)

is nonnegative on the entire space. Then it is worthy to catalogue our simple observation. An inhomogeneous quadratic form \( f_\xi(x) = x^T Ax + 2b^T(\lambda)x + c(\lambda) - \xi \)
is nonnegative everywhere if and only if \((\lambda, \xi)\) satisfy the following linear matrix inequality:

\[
\begin{bmatrix}
    c(\lambda) - \xi & b^T(\lambda) \\
    b(\lambda) & A
\end{bmatrix} \geq 0
\]  

(35)

where Equation (35) means the left matrix is positive semidefinite.

Expanding linear matrix inequality as

\[
\begin{bmatrix}
    \sum_{i=1}^{N} \lambda_i(-c_i) - \xi & b + \sum_{i=1}^{N} \frac{1}{2} \lambda_i A_i \\
    b + \sum_{i=1}^{N} \frac{1}{2} \lambda_i A_i & A
\end{bmatrix} \geq 0
\]  

(36)

Then \(\xi\) is a lower bound for the optimal value.

### 5.2. Conic quadratic program

Consider that quadratic program problem again, in real applications, the data \(A, b, A_i, c_i, i = 1, 2, \ldots N\) are not always known exactly, what is typically known is a domain \(\Omega\) in the space of data – an uncertainty set, which for sure contains the actual data. There are cases in reality where in spite of this data uncertainty, our decision \(x\) must satisfy the actual constraints, whether we know them or not.

If indeed all we know about the data is that they belong to a given set \(\Omega\), but we still have to satisfy the actual constraints, the only way to meet the requirements is to restrict ourselves to robust feasible solutions – those satisfying all possible realizations of the uncertain constraints, i.e. vectors \(x\) such that

\[
\begin{bmatrix}
    A_1 \\
    A_2 \\
    \vdots \\
    A_N
\end{bmatrix} x -
\begin{bmatrix}
    c_1 \\
    c_2 \\
    \vdots \\
    c_N
\end{bmatrix} \leq 0, \quad \forall (A_i, c_i)_{i=1}^{N}
\]  

(37)

such that \(\exists A\) and \(b : (A_i, c_i, A, b) \in \Omega, \Omega\) is one given set.

In order to choose among these robust feasible solutions the best possible, we should decide how to aggregate various realizations of the objective into a single characteristic. In other words, the robust counterpart of quadratic program problem (31) is the following optimization problem:

\[
\min_{t, x} tx^T A x + 2b^T x \leq t, \quad \forall (A_i, c_i, A, b)_{i=1}^{N} \in \Omega
\]  

(38)

Note that Equation (38) is a usual optimization problem. As we see in a while, in many cases, it is reasonable to specify the uncertainty set \(\Omega\) as an ellipsoid, i.e. the image of the unit Euclidean ball under an affine mapping or as a conic quadratic program. Then in this case, the robust counterpart of an uncertain quadratic program problem is an explicit conic quadratic program. Thus robust quadratic program with conic quadratic uncertainty sets can be viewed as a generic source of conic quadratic problem.

Observing the robust counterpart of an uncertain quadratic program.

\[
\{\min_{x} x^T A x + 2b^T x, A x \leq c_i, i = 1, 2 \ldots N\}
\]

\[
(A_i, c_i, A, b)_{i=1}^{N} \in \Omega
\]

(39)

In the case of a simple ellipsoidal uncertain, assume that uncertainty set is

\[
\Omega = \{(A_1, c_1, \ldots , A_N, c_N, A, b) : \exists (e_i, e_i^T e_i \leq 1)_{i=1}^{N}:
\]

\[
\begin{bmatrix}
    A_i \\
    c_i
\end{bmatrix}
+ P_i e_i, \quad i = 1, 2 \ldots N
\]

\[
\begin{bmatrix}
    A \\
    b
\end{bmatrix}
+ e_0\}
\]

(40)

where \(A_0, b_0, A^*_i, c^*_i\) are the nominal data and \(P_i e_i, i = 1, 2 \ldots N\), represent the data perturbations, the restrictions \(e_i^T e_i \leq 1\) enforce these perturbations to vary in ellipsoids.

In order to realize that the robust counterpart of our uncertain quadratic program problem is a conic quadratic problem, note that \(x\) is robust feasible if and only if for every \(i = 1, 2 \ldots N\), we have

\[
\max_{e_i | e_i^T e_i \leq 1} [A_i(e) x - c_i(e) :
\]

\[
\begin{bmatrix}
    A_i(e) \\
    c_i(e)
\end{bmatrix}
+ P_i e_i
\]

\[
= A^*_i x - c^*_i + \max_{e_i | e_i^T e_i \leq 1} x_i^T P_i T [\begin{bmatrix}
    x \\
    -1
\end{bmatrix}]
\]

\[
= A^*_i x - c^*_i + \left\| P_i T [\begin{bmatrix}
    x \\
    -1
\end{bmatrix}] \right\|_2 \leq 0
\]

(41)

Thus \(x\) is robust feasible if and only if it satisfies the system of

\[
\left\| P_i T [\begin{bmatrix}
    x \\
    -1
\end{bmatrix}] \right\|_2 \leq c^*_i - A^*_i x, \quad i = 1, 2 \ldots N
\]

(42)

Similarly, a point \((x, t)\) satisfies all realizations of the inequality \(x^T A x + 2b^T x \leq t\) by our ellipsoidal uncertainty set \(\Omega\) if and only if

\[
x^T A x + x^T P A e_0 x + 2b x + 2P e_0 x \leq t
\]

(43)
Thus the robust counterpart becomes one conic quadratic program
\[
\min_{t,x} \left\{ t : \|PAx\|_2 + \|PBx\|_2 \leq -x^TA_xx - 2b_xx + t; \right. \\
\left. \left\| P^T \begin{bmatrix} x \\ 1 \end{bmatrix} \right\|_2 \leq c^*_i - A^*_ix_i, \ i = 1, 2 \ldots N \right\}
\]
(44)

6. Simulation example

Now we propose one simulation example to illustrate the nature of the above results. Consider one problem of multi UAVs formation cooperative. As we know that multi UAVs formation cooperative depends on the transmission of information and sharing in case of implementing the task. Task coupling and information sharing make the formation cooperative as a distributed networked intelligence system, where each UAV is regarded as one knot and one wireless communication network is proposed to realize the communication among all UAVs. Based on this distributed networked intelligence system, each UAV can perform the monitoring task according to different locations for mission planning. One special architecture of monitoring mission for multi UAVs formation cooperative reconnaissance is plotted in Figure 1, where many cooperative modules are seen explicitly.

By using this network module, there are many types of function modules for each UAV, for example, cooperative mission planning, cooperative trajectory planning, tracking, attacking and searching, etc. Network module sends control requirement to cooperative mission planning, so that each UAV is controlled by its network goal. All of these function modules will communicate with their neighbouring UAVs to share information resources. This architecture in Figure 1 can be simplified to Figure 2, where the basic internal structure of the function module is given to achieve the cooperative mission planning.

Further from flight control theory, we see that UAV reconnaissance platform acquires a frame of image, coinciding with ground target and landscape. Two cases exist for extracting target location information from the received image. The ground target positioning process is plotted in Figure 3, where the primary target is in the centre of the field of the airborne optoelectronic platform, and the secondary target is not in the centre of the image. As each UAV is regarded as one knot and one wireless communication network is proposed to realize the communication among all UAVs in Figure 4, so there are many communications among these UAVs, and these UAVs can be deemed as some subsystems.

**Figure 1.** Architecture of monitoring mission for multi UAVs formation cooperative reconnaissance.
The formation includes three UAVs: one radar jamming UAV, one missile jamming UAV and one investigation UAV. The initial positions of three UAVs are all located at starting point coordinates (0, 0) and the terminate positions are concentrated on coordinates (700, 700). The vector of the flight maximum speed, minimum speed, and the speed deviation is
\[
[80 \ 15 \ \pm 5]
\]  \hspace{2cm} (45)

Surrounding battlefield environment contains radar threat, missile threat and antiaircraft positions threat. The deployment coordinates of the radar threat are (300, 300), and the deployment coordinates of the missile threat are (250, 200). The region of the artillery positions threat is a rectangle range with a height of 300 and a width of 300. This rectangle range belongs to the no-fly zone.

When sample period is \( T \), matrices \( A_{ij} \) and \( B_{ij} \) are defined as follows respectively
\[
A_{ij} = \begin{bmatrix}
1 & T & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & T & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & T \\
0 & 0 & 0 & 0 & 0 & 0 
\end{bmatrix},
\]  
\[
B_{ij} = \begin{bmatrix}
\frac{T^2}{2} & 0 & 0 \\
\frac{T}{2} & 0 & 0 \\
0 & \frac{T^2}{2} & 0 \\
0 & 0 & \frac{T^2}{2} \\
0 & 0 & 0 \\
0 & 0 & \frac{T^2}{2} 
\end{bmatrix}
\]  \hspace{2cm} (46)

In simulation environment, the input signal is the excited signal chosen by user, output is measured from the point set collected by the accelerator, the number of sampled points is set \( N = 4096 \). The output and input of the 4096 sampled data are divided into 4 equal data blocks; each data block contains 1024 sample data.
The weighted factor for each UAV is
\[ Q = \text{diag}\{10, 10\} \]
The weighted matrix in the cost function is
\[ R = \text{diag}\{1, 10\} \]
The discrete time sampling period is taken as
\[ T = 0.05 \text{ s} \]

The simulation environment under windows system.

The simulation scenario includes three UAVs which constitute wedge formation in the two-dimensional plane. The initial state of each UAV is respectively
\[(500, 500), \ (500, 5000) \ (2000, 5000)\]
In the final formation form, the displacements of each UAV relative to the confluence point are
\[(-100, 100), \ (-100, 100) \ (0, 0)\]
The optimal convergence point is \((3200, 3500)\). The formation speed at the convergence point is \((200, 0)\). The maximum acceleration speed of each UAV is limited and given as \(35 \text{ m/s}^2\). The maximum speed is \(250 \text{ m/s}\), the total execution time is \(30 \text{ s}\). Based on the dynamic programming strategy, each UAV can find the optimal confluent point which makes the energy consumption lower in the formation process. Three different colour curves in Figure 5 represent the best track trajectories after the optimal confluent point is determined by optimization. The iteration curve about the objection criterion function as the iteration steps is given in Figure 6. From Figure 6, we see that when the number of the iteration step is 600

**Figure 4.** Communications among each subsystem.

**Figure 5.** The optimization results of each UAV.
times, the whole algorithm will be terminated and the stable convergent is guaranteed.

7. Conclusion

This paper connects system identification, model predictive control and convex optimization theory to construct data-driven model predictive control. The combination of system identification and model predictive control is to derive the actual output value for cost function, and convex optimization theory is used to do some robustness analysis for our reformulated quadratic program problem with inequality constraints, corresponding to the optimization problem within model predictive control. As this is our primarily analysis for data-driven model predictive control, that about how to merge game theory and data-driven model predictive control is our next ongoing work.

Acknowledgements

This work is partially supported by the Grants from the Foundation of Jiangxi University of Science and Technology (No. 20202BABL202009). We are grateful to Professor Eduardo F Camacho for his warm invitation in his control lab at the University of Seville, Seville, Spain. Furthermore, due to the sudden Covid 2019, the first author cancels the visiting to Professor Olivier Sename in Univ. Grenoble-Alpes, thanks for his warm invitation in his adaptive control lab.

Data availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work is partially supported by the Grants from the Foundation of Jiangxi University of Science and Technology (No. 20202BABL202009).

References

Alamo, T., Tempo, R., & Camacho, E. F. (2009). Randomized strategy for probabilistic solution of uncertain feasibility and optimization problems. *IEEE Transactions on Automatic Control, 54*(11), 2545–2559. https://doi.org/10.1109/TAC.2009.2031207

Bertsimas, D., & Goyal, V. (2012). On the power and limitations of affine policies in two stage adaptive optimization. *Mathematical Programming, 134*(2), 491–531. https://doi.org/10.1007/s10107-011-0444-4

Blackmore, L., Ono, M., & Williams, B. C. (2011). Chance constrained optimal path planning with obstacles. *IEEE Transactions on Robotics, 27*(6), 1080–1094. https://doi.org/10.1109/TRO.2011.2161160

Calafiore, G. C. (2010). Random convex programs. *SIAM Journal on Optimization, 20*(6), 3427–3464. https://doi.org/10.1137/090773490

Callaway, D., & Hiskens, I. (2011). Achieving controllability of electric loads. *Proceedings of the IEEE, 99*(1), 184–199. https://doi.org/10.1109/JPROC.2010.2081652

Campi, M. C., & Garatti, S. (2016). Wait and judge scenario optimization. *Automatica, 50*(12), 3019–3029. https://doi.org/10.1017/S10107-016-1056-9

Campi, M. C., Garatti, S., & Prandini, M. (2009). The scenario approach for systems and control design. *Annual Reviews in Control, 33*(2), 149–157. https://doi.org/10.1016/j.arcontrol.2009.07.001

Farina, M., Giulioni, L., & Scattolini, R. (2016). Stochastic linear model predictive control with chance constraints: A review. *Journal of Process Control, 44*(2), 53–67. https://doi.org/10.1016/j.jprocont.2016.03.005

Garatti, S., & Campi, M. C. (2013). Modulating robustness in control design: principles and algorithm. *IEEE Control Systems Magazine, 33*(2), 36–51. https://doi.org/10.1109/MCS.2012.2234964

Zhang, X., Grammatico, S., Schildbach, G., Goulart, P., & Lygeros, J. (2015). On the sample size of random convex programs with structured dependences on the uncertainty. *Automatica, 60*, 182–188. https://doi.org/10.1016/j.automatica.2015.07.013

Zhang, X., Kamgarpour, M., Georgiou, A., Goulart, P., & Lygeros, J. (2017). Robust optimal control with adjustable uncertainty sets. *Automatica, 75*, 249–259. https://doi.org/10.1016/j.automatica.2016.09.016