Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Overcoming Singularity and Degeneracy in Neighboring Extremal Solutions of Discrete-Time Optimal Control Problem with Mixed Input-State Constraints ⋆

Reza Ghaemi *, **Jing Sun** **, **Ilya Kolmanovsky** ***

* *Department of Electrical Engineering and Computer Science at the University of Michigan, Ann Arbor, MI 48109; email:ghaemi@umich.edu*
** *Department of Naval Architecture and Marine Engineering at the University of Michigan, Ann Arbor, MI 48109; email:jingsun@umich.edu*
*** *Ford Motor Company, 2101 Village Road, Dearborn, MI 48124, email: ikolmano@ford.com*

Abstract: Neighboring Extremal Optimal approach is effective in solving optimal control problems through approximation. Under certain conditions, a matrix involved in the calculation of optimal control approximation can become singular, leading to a technical difficulty in the application of the approach. These situations may include the cases when a constraint depends only on the states but not the inputs, or the cases when the inequality constraints outnumber the inputs. In this paper, we propose a solution that can circumvent this technical difficulty. First, by back-propagating the state constraints, we show that input-independent constraints can be recast as the state-input constraints to avoid the matrix singularity. The back-propagation, however, might lead to another problem of "degeneracy," where a back-propagated constraint is imposed on the initial state, so that no feasible neighboring extremal solution exists for the problem. In the latter case, a linear programming approach is proposed to deal with this degeneracy. A ship maneuvering control problem is used in the paper to illustrate the singularity and degeneracy issues, and to elucidate the mechanics of the proposed scheme.

1. INTRODUCTION

Optimization-based control refers to a control design in which control decisions are made by solving optimization problems on-line. One prominent example of the optimization-based control is the model predictive control (MPC), which has enjoyed widespread success in the process industry (Binder et.al (2001), Qin and Badgwill, (1997)). The flexible and intuitive formulation, together with the capability of incorporating state and input constraints, has made the approach very attractive to a broad class of applications, such as those in aerospace, automotive and marine industry. For systems whose dynamics are fast and/or whose computational resources are limited, however, the practicality of the optimization-based control is often challenged. The challenges arise for several reasons: First, the optimization has to be repeated at each sampling instant when the control needs to be updated. Second, for nonlinear systems with constraints, or even for linear systems with constraints, analytical solutions to the optimization problems are often unattainable. Research efforts in making MPC or other optimization-based control feasible for applications with fast dynamics and limited computational resources have been successful and recent advances included explicit MPC (Johansen (2002), Bemporad (2002)) and model reduction (Dufour (2003)), among the others.

Another venue for making MPC or other optimization-based control appealing in applications is to improve the efficiency of numerical optimization algorithms while exploiting special features of underlying problem. For a given application, by noting that a similar constrained optimization problem is to be solved repeatedly as the states and inputs evolve, one can leverage the neighboring extremal approach to compute the solution at the next sampling instant starting from the solution at the current time instant. This idea has been explored in (Ghaemi et. al. (2007a)). In (Ghaemi et. al. (2007b)) this perturbation analysis based approach is integrated with the sequential quadratic programming (SQP) to deal with cases when large state variations are involved.

For the neighboring extremal solution described in (Ghaemi et. al. (2007b)), several cases are identified where the proposed algorithm cannot be applied due to the singularity of a matrix involved in the neighboring extremal control calculation. These cases include those when some constraints depend only on the states but not on the inputs, or those when the active linearized constraints outnumber the inputs.

⋆ This work is supported in part by NSF ECS-0501284 and ONR N00014-05-1-0537.

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.3684

In this paper, we propose an algorithm that circumvents this technical difficulty. The mitigating strategy has several key ingredients: one involves the back-propagation of the constraints to avoid singularity issues, and the other relies on a linear programming algorithm to avoid the degeneracy issue. For the purpose of clear exposition, the issues and concepts behind the algorithm will be elucidated using ship maneuvering control as a numerical example.

2. NEIGHBORING EXTREMAL CONTROL

In this section we discuss the neighboring extremal control approach for discrete-time systems with joint input and state inequality constraints, proposed by (Ghaemi et. al. (2007a)). Consider the problem of minimizing a cost function,

$$J[u] = \sum_{k=0}^{N-1} L(x(k), u(k)) + \Phi(x(N)), \tag{1}$$

over all feasible control sequences $u : [0, N-1] \to \mathbb{R}^m$ and all state vectors $x : [0, N] \to \mathbb{R}^n$ subject to the following constraints:

$$x(k+1) = f(x(k), u(k)); \quad f : \mathbb{R}^{n+m} \to \mathbb{R}^n \tag{2}$$

$$x(0) = x_0; \ x_0 \in \mathbb{R}^n \tag{3}$$

$$C(x(k), u(k)) \leq 0, \ C : \mathbb{R}^{n+m} \to \mathbb{R}^l. \tag{4}$$

The neighboring extremal solution to the problem when the initial state is perturbed from $x(0) + \delta x(0)$ is derived in (Ghaemi et. al. (2007b)) and it is described as follows:

Let $x^o(k)$, $u^o(k)$, referred to as the nominal solution, be the state and control trajectories corresponding to the optimal solution in the problem of minimizing (1) subject to (2)-(4) with the initial condition $x(0)$. Let $C^a(k)$ be a vector consisting of the constraints that are active at the time instant k [1], $\mu(k)$ be the corresponding Lagrange variable, and $\lambda(\cdot)$ be the sequence of co-states associated with the dynamics of the system. The Hamiltonian function can be defined as follows:

$$\begin{aligned} H(k) &= L(x(k), u(k)) + \lambda(k+1)^T f(x(k), u(k)) \\ &\quad + \mu(k)^T C^a(x(k), u(k)). \end{aligned} \tag{5}$$

As shown in (Ghaemi et. al. (2007b)), if a perturbation $\delta x(0)$ in the initial state $x(0)$ does not change the activity status of the constraints, then the corresponding optimal solution to the problem defined by the cost function (1) and constraints (2), (4) and initial state $x(0) = x_0 + \delta x(0)$ can be approximated as $x^o(k) + \delta x(k)$ and $u^o(k) + \delta u(k)$, $k \in [0, N]$, provided

$$Z_{uu}(k) \succ 0 \text{ for } k \in [0, N] \tag{6}$$

for the nominal solution, where [2]

$$\begin{aligned} Z_{uu}(k) &= H_{uu}(k) + f_u^T(k) S(k+1) f_u(k), \\ Z_{ux}(k) &= Z_{xu}(k)^T = H_{ux}(k) + f_u^T(k) S(k+1) f_x(k), \\ Z_{xx}(k) &= H_{xx}(k) + f_x^T(k) S(k+1) f_x(k), \end{aligned} \tag{7}$$

[1] Note that the dimension of $C^a(k)$ can vary for different k. It is an empty vector (considered to be full rank) if no constraint is active at the time instant k.

[2] Subsequently, H_{uu}, H_{ux}, H_{xx}, Φ_{xx}, etc., denote the partial derivatives with respect to x and/or u, with the exception for Z_{uu}, Z_{ux}, Z_{xx}, which are defined by (7).

and $S(k)$ in equation (7) is given by:

$$\begin{aligned} S(k) &= Z_{xx}(k) - [Z_{xu}(k) \ C_x^T(k)] K_0(k) \begin{bmatrix} Z_{ux}(k) \\ C_x(k) \end{bmatrix}, \\ S(N) &= \Phi_{xx}(N), \end{aligned} \tag{8}$$

$$K_0(k) = \begin{bmatrix} Z_{uu}(k) & C_u^{aT}(k) \\ C_u^a(k) & 0 \end{bmatrix}^{-1}. \tag{9}$$

Moreover, the following explicit relation between state and input variations is satisfied for the perturbed solution:

$$\delta u(k) = K^*(k) \delta x(k), \tag{10}$$

$$K^*(k) = -[I \ 0] K_0(k) \begin{bmatrix} Z_{ux}(k) \\ C_x^a(k) \end{bmatrix}. \tag{11}$$

Remark 1. Note that in order to calculate the neighboring extremal solution, one also needs to calculate nominal trajectories of $\mu(k)$ and $\lambda(k)$.

Given that the matrix $Z_{uu}(\cdot)$ is positive definite over the entire horizon, according to assumption (6), one can easily verify that the matrix $K_o(k)$ in (9) is well defined as long as $C_u^a(k)$ is full rank for $k = 0, \cdots, N-1$. If $C_u^a(k)$ is not full rank at some time instant k, then

$$G(k) := \begin{bmatrix} Z_{uu}(k) & C_u^{aT}(k) \\ C_u^a(k) & 0 \end{bmatrix}$$

becomes singular and the proposed algorithm fails because the inverse in (9) is not defined. The singularity happens if the matrix C_u is not row independent. Two special cases of such situation are when $C_u = 0$, or when the number of active inequality constraints at time k is greater than the number of inputs, i.e, C_u^a has more rows than columns.

In the next section, the ship steering problem will be introduced as an illustrating example, where both cases highlighted above will be illuminated.

3. A NUMERICAL EXAMPLE

In a ship maneuvering problem, the objective is to steer a ship to a desired location while avoiding an obstacle, which could be an oil rig or another ship. The following ship model, taken out from Casado et. al, (2001), is used for numerical simulation:

$$\begin{aligned} \dot{x}_1 &= x_5 cos(x_3) - (r_1 x_4 + r_3 x_4^3) sin(x_3), \\ \dot{x}_2 &= x_5 sin(x_3) + (r_1 x_4 + r_3 x_4^3) cos(x_3), \\ \dot{x}_3 &= x_4, \\ \dot{x}_4 &= -a x_4 - b x_4^3 + c u_r, \\ \dot{x}_5 &= -f x_5 - W x_4^2 + u_t, \end{aligned} \tag{12}$$

where x_1 and x_2 are the ship's position (in nautical miles (nm)) in the $X_1 - X_2$ plane, x_3 is the heading angle (in radians (rad)), x_4 is the yaw rate (rad/min), and x_5 is the forward velocity (nm/min). The two control inputs are: u_r, the rudder angle (rad), and u_t, the propeller's thrust (nm/min^2).

The ship has a maximum speed of .25 nm/min = 15 knots for a maximum thrust of 0.215 nm/min^2. For the maximal rudder angle of 35^o, the stationary rate of turn is 1^o/sec.

The control objective is to steer the ship from any initial condition to a neighborhood around the origin (described by a circle with a radius $r_0 = 0.1(nm)$) while minimizing energy consumption. Moreover, there is an obstacle designated by a circle centered at $(x_1, x_2) = (1.5, 0)$ with radius $\bar{r} = 2.5(nm)$ so that the following inequality constraint, which depends only on the state needs to be enforced,

$$(x_1 - 1.5)^2 + x_2^2 \geq (0.25)^2. \tag{13}$$

The objective of energy consumption minimization, coupled with the need to satisfy hard constraints due to actuator saturation and the presence of an obstacle, naturally suggests the model predictive control (MPC) as a strategy. At each time instant, the calculation of the control signal in MPC involves solving an optimization problem of minimizing the cost

$$\min_{u(\cdot),x(\cdot)} \sum_{i=0}^{N-1} (0.1u_r(k)^2 + 10u_t(k)^{3/2}) + 2000(x_1(k)^2 + x_2(k)^2) \tag{14}$$

subject to constraints

$$\begin{aligned} &C_1(k) = u_r(k)^2 \leq 0.61^2 \\ &C_2(k) = (u_t(k) - 0.125)^2 \leq 0.125^2 \\ &\bar{C}(k) = (x_1(k) - 1.5)^2 + x_2(k)^2 \geq (0.25)^2 \\ &x(0) = \text{observed state} \\ &x(k+1) = f(x(k), u(k)) \quad u = [u_r\ u_t]^T, \end{aligned} \tag{15}$$

where the first two inequalities are the control limits, and the discrete-time model of the system is derived by discretizing equation (12), using Euler approximation. According to the MPC strategy, the first element in the optimized control sequence $u^*(\cdot)$ will be applied as control signal. Note that optimization problem (14)-(15) has to be solved repeatedly for each updated observed state. Once a solution for (14)-(15) is obtained for the state $x(i)$ observed at $t = iT$ and assumed as an initial state in (14)-(15), then for the next sample time $t = (i+1)T$ with observed state $x(i+1)$, one can use the neighboring extremal method to approximate the new solution by assuming $\delta x(0) = x(i+1) - x(i)$; thereby reducing the impact of the computing delay.

Remark 2. The neighboring extremal method described in Section 2 assumes that the constraint activity status at each time k does not change after the initial condition perturbation. This problem is resolved in (Ghaemi et. al. (2007a)), where an algorithm is proposed to handle perturbations that are large enough to cause activity status changes for some constraints. Figure 1 shows the trajectory of the ship maneuvering problem when the neighboring extremal method in (Ghaemi et. al. (2007a)) is applied, where the problems with the singular matrix $G(k)$ are resolved using the algorithm proposed in this paper, as elaborated in the subsequent sections.

4. AVOIDING SINGULARITY IN THE NEIGHBORING EXTREMAL SOLUTION

In this section, we propose an approach which permits the neighboring extremal solution to be applied without requiring $C_u^a(k)$ to be full rank. Note that in the spirit of the neighboring extremal solution, the linear constraints

$$C_u^a(k)\delta u(k) + C_x^a(k)\delta x(k) = 0 \tag{16}$$

should be satisfied, where $C^a(k)$ denotes all active constraints at the time k. When $C_u^a(k)$ has dependent rows, it can be transformed into the following form

$$\begin{bmatrix} \tilde{C}_u(k) \\ 0 \end{bmatrix}$$

for some $\tilde{C}_u(k)$ with independent rows. Therefore, equation (16) can be decomposed into

$$\tilde{C}_u(k)\delta u(k) + \tilde{C}_x(k)\delta x(k) = 0, \tag{17}$$

$$\hat{C}_x(k)\delta x(k) = 0, \tag{18}$$

for appropriately defined $\tilde{C}_x(k)$ and $\hat{C}_x(k)$. Using the linearized version of (2), namely

$$\delta x(k+1) = f_x(k)\delta x(k) + f_u(k)\delta u(k) \tag{19}$$

for $k > 0$, (18) can be rewritten as

$$\hat{C}_x(k)\left(f_x(k-1)\delta x(k-1) + f_u(k-1)\delta u(k-1)\right) = 0. \tag{20}$$

Therefore, we can effectively replace the constraints (16) by (17), and the remaining constraints (18) are back-propagated to the time instant $k-1$, so that constraints are now imposed on $\delta x(k-1)$ and $\delta u(k-1)$. Now we can redefine the matrix $K_0(k)$ as

$$K_0(k) = \begin{bmatrix} Z_{uu}(k) & \tilde{C}_u^T(k) \\ \tilde{C}_u(k) & 0 \end{bmatrix}^{-1}, \tag{21}$$

to avoid singularity in (9). This technique, which mitigates the singularity by redefining the constraints at time k and shifting other state-only constraints to time $k-1$, is referred to as constraint back-propagation in this paper. Since the back-propagated constraints (20) have to be enforced along with other active constraints, at time $k-1$, the resulting active constraints may outnumber control inputs at the time $k-1$. If this does indeed happen, then the same technique has to be repeated until at some point $k-j$ the back-propagated constraints can be absorbed by the matrix $C^a(k-j)$ without causing any singularity in $G(k-j)$.

To illustrate the concept of constraint back-propagation, we consider the ship steering problem discussed in Section 3, where the problem (14)-(15) is solved at each time instant. The ship trajectory is shown in Figure 1. At $i = 69$, which is highlighted by point A on the figure, the state constraint (13) becomes active at $k = 10$ (i.e., 10-steps ahead for the predicted trajectory).

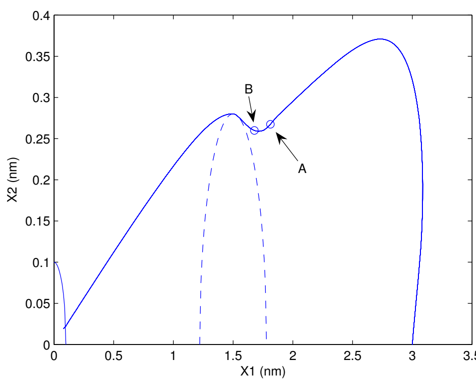

Fig. 1. Ship trajectory and the singularity points.

At $k = 10$, the constraint $\bar{C}(10) \leq 0$ being active leads to the following constraint on the state variation $\delta x(10)$ (see also Table I):

$$[-0.09 \ -0.26 \ 0 \ 0 \ 0]\,\delta x(10) = 0. \tag{22}$$

This constraint can be back-propagated to $k = 9$ as

$$[-0.09 \ -0.26 \ 0.0072 \ -0.001 \ 0.01]\,\delta x(9) = 0 \tag{23}$$

using the state equation. Note that the system has relative degree equal to 2, therefore $\delta u(9)$ does not appear in (23) and the singularity remains to be the problem. Now using (20) for $k = 9$, (23) can be back-propagated to $k = 8$ as:

$$\begin{bmatrix} -0.0003 \\ 0.0010 \end{bmatrix}^T \delta u(8) + \begin{bmatrix} -0.0962 \\ -0.2630 \\ 0.0145 \\ -0.0012 \\ 0.0184 \end{bmatrix}^T \delta x(8) = 0, \tag{24}$$

which, when combined with the original constraint at $k = 8$

$$\begin{bmatrix} 0.61 & 0 \\ 0 & 0.1 \end{bmatrix} \delta u(8) = 0, \tag{25}$$

leads to the singularity of matrix $K_0(8)$.

Note that for $k = 8$, we can combine (24) and (25) as

$$\begin{bmatrix} 0.61 & 0 \\ 0 & 0.1 \\ -0.003 & 0.001 \end{bmatrix} \delta u(8) + \begin{bmatrix} C_x^a(8) \\ v_8 \end{bmatrix} \delta x(8) = 0, \tag{26}$$

where $C_x^a(8) = 0_{2\times 5}$ and

$$v_8 = [-0.0962 \ -0.2630 \ 0.0145 \ -0.0012 \ 0.0184]$$

is the coefficient vector corresponding to $\delta x(8)$ in (24). The equation (26), through matrix row operation, can be expressed as

$$\begin{bmatrix} 0.61 & 0 \\ 0 & 0.1 \end{bmatrix} \delta u(8) = 0 \ \text{ and } v_8 \delta x(8) = 0. \tag{27}$$

Now continuing with constraint back-propagation for the second equation in (27), we have Table I which gives all the constraints recast after each step of back-propagation. It can be seen that after 5 steps of back-propagation, at $k = 6$, the singularity problem can be avoided and the neighboring extremal solution can be defined and applied.

5. CONSTRAINT BACK-PROPAGATION ALGORITHM

In this section we formalize and generalize the back-propagation method which has been described conceptually in the previous section. Let us consider the optimization problem of minimizing the cost (1) subject to the following constraints:

$$x(k+1) = f(x(k), u(k)); \quad f : \mathbb{R}^{n+m} \to \mathbb{R}^n, \tag{28}$$

$$x(0) = x_0; \ x_0 \in \mathbb{R}^n, \tag{29}$$

$$C(x(k), u(k)) \leq 0, \ C : \mathbb{R}^{n+m} \to \mathbb{R}^l, \tag{30}$$

$$\bar{C}(x(k)) \leq 0 \ \ \bar{C} : \mathbb{R}^n \to \mathbb{R}^{\bar{l}}, \tag{31}$$

where C and $\bar{C}$ denote the mixed state-input constraints and state-only constraints, respectively. This set of constraints includes state only inequality constraint (31), in contrast to (4).

In order to describe the neighboring extremal solution to this problem, we first introduce the following notations. Let $x(k)$, $u(k)$, $k \in [0, N]$ be referred to as the nominal solution for the state and control corresponding to the optimal solution in the problem of minimizing (1) subject to the constraints (28)-(31). With $\lambda(\cdot)$ being defined as in Section 2, the Hamiltonian function can be defined as follows:

$$\begin{aligned} H(k) = & \ L(x(k), u(k)) + \lambda(k+1)^T f(x(k), u(k)) \\ & + \mu(k)^T C^a(x(k), u(k)) + \bar{\mu}(k)^T \bar{C}^a(x(k)), \end{aligned} \tag{32}$$

where $\mu(\cdot)$ and $\bar{\mu}(\cdot)$ are Lagrange multipliers associated with the active parts of constraints (30) and (31), respectively. Now we define matrix sequences $\tilde{C}_u(\cdot)$, $\tilde{C}_x(\cdot)$ and $S(\cdot)$ using the following backward recursive equations. Let

$$\begin{aligned} \hat{C}_x(N) &:= \bar{C}_x^a(x(N)), \\ S(N) &:= \Phi_{xx}(N) + \frac{\partial}{\partial x}(\bar{C}_x^T(x(N))\bar{\mu}(N)), \end{aligned} \tag{33}$$

and at the time instant k we define

$$\begin{aligned} C_{aug}(k) &:= \begin{bmatrix} C_u^a(k) \\ \hat{C}_x(k+1) f_u(k) \end{bmatrix}, \\ \tilde{r}_k &:= rank(C_{aug}(k)). \end{aligned} \tag{34}$$

At each time instant k, there is a matrix $P(k)$ that transforms matrix $C_{aug}(k)$ into the following form

$$P(k) \begin{bmatrix} C_u(k) \\ \hat{C}_x(k+1) f_u(k) \end{bmatrix} = \begin{bmatrix} \tilde{C}_u(k) \\ 0 \end{bmatrix}, \tag{35}$$

with $\tilde{C}_u(k) \in \mathbb{R}^{\tilde{r}_k \times m}$ having independent rows. If $C_{aug}(k)$ is full row rank, then $P(k) = I$ and $\tilde{C}_u(k) = C_{aug}(k)$.

By defining

$$\Gamma(k) := \begin{bmatrix} P(k) \begin{bmatrix} C_x^a(k) \\ \hat{C}_x(k+1) f_x(k) \end{bmatrix} \\ \bar{C}_x^a(k) \end{bmatrix},$$

and assuming that γ_k is the number of rows of matrix $\Gamma(k)$, we can define

$$\begin{aligned} \tilde{C}_x(k) &:= [I_{\tilde{r}_k \times \tilde{r}_k} \ 0_{\tilde{r}_k \times (\gamma_k - \tilde{r}_k)}]\Gamma(k) \in \mathbb{R}^{\tilde{r}_k \times m}, \\ \hat{C}_x(k) &:= [0_{(\gamma_k - \tilde{r}_k)\times \tilde{r}_k} \ I_{(\gamma_k - \tilde{r}_k)\times(\gamma_k - \tilde{r}_k)}]\Gamma(k) \in \mathbb{R}^{(\gamma_k - \tilde{r}_k)\times m}. \end{aligned} \tag{36}$$

Having $Z_{uu}(\cdot)$, $Z_{ux}(\cdot)$ and $Z_{xx}(\cdot)$ defined in (7), the matrix $S(k)$ can be defined as follows

$$S(k) = Z_{xx}(k) - [Z_{xu}(k) \ \tilde{C}_x^T(k)] K_0(k) \begin{bmatrix} Z_{ux}(k) \\ \tilde{C}_x(k) \end{bmatrix}, \tag{37}$$

where

$$K_0(k) = \begin{bmatrix} Z_{uu}(k) & \tilde{C}_u(k)^T \\ \tilde{C}_u(k) & 0 \end{bmatrix}^{-1}.$$

Using equation (33) as an initial condition for a backward run, we apply equations (35), (36) and (37) to calculate matrix sequences $Z_{uu}(\cdot)$, $Z_{ux}(\cdot)$, $Z_{xx}(\cdot)$, $\tilde{C}_u(\cdot)$, $\tilde{C}_x(\cdot)$ and $P(\cdot)$. Having the above matrix sequences calculated, we introduce the following theorem which gives a sufficient condition for existence of the neighboring extremal solution to the optimal control problem with perturbed initial state. The theorem is followed by a corollary which gives the neighboring extremal solution.

Theorem 3. If $rank(\hat{C}_x(0)) = 0$ then a sufficient condition for the existence of the neighboring extremal control subject to the inequality constraints and initial state perturbation $\delta x(0)$ is

$$Z_{uu}(k) \succ 0 \text{ for } k \in [0, N-1]. \tag{38}$$

Table I: The content of matrices $C_u^a(\cdot)$, $\bar{C}_x^a(\cdot)$ and $\hat{C}_x(\cdot)$ for constraint (13).

k	0	1	2	3	4	5	6	7	8	9	10
$C_u^a(\cdot)$	$\begin{bmatrix} -0.61 & 0 \\ 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.61 & 0 \\ 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0.61 & 0 \\ 0 & 0.1 \end{bmatrix}$	$\begin{bmatrix} 0 & -0.1 \end{bmatrix}$	
$\bar{C}_x^a(\cdot)$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\begin{bmatrix} -0.09 \\ -0.26 \\ 0 \\ 0 \\ 0 \end{bmatrix}$
$\hat{C}_x(\cdot)^T$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\begin{bmatrix} -0.09 \\ -0.263 \\ 0.022 \\ -0.0006 \\ 0.026 \end{bmatrix}$	$\begin{bmatrix} -0.09 \\ -0.26 \\ 0.01 \\ -0.0012 \\ 0.018 \end{bmatrix}$	$\begin{bmatrix} -0.09 \\ -0.26 \\ 0.0072 \\ -0.001 \\ 0.01 \end{bmatrix}$	$\begin{bmatrix} -0.09 \\ -0.26 \\ 0 \\ 0 \\ 0 \end{bmatrix}$

Remark 4. Condition (38) guarantees the convexity of the quadratic programming problem resulting from the second order variational analysis, which is performed to calculate the neighboring extremal solution.

Remark 5. The condition, $rank(\hat{C}_x(0)) = 0$, implies that the back-propagation will not result in a constraint on the initial state variation $\delta x(0)$, which is not an optimization variable.

Corollary 6. If a perturbation $\delta x(0)$ in the initial state $x(0)$ does not change the activity status of the constraints and

$$Z_{uu}(k) \succ 0 \text{ for } k \in [0, N-1], \tag{39}$$

then the corresponding optimal solution to the problem defined by the cost function (1) and constraints (28)-(31) and initial state $x(0) = x_0 + \delta x(0)$ can be approximated as $x(k) + \delta x(k)$ and $u(k) + \delta u(k)$, $k \in [0, N]$ where

$$\delta u(k) = K^*(k)\delta x(k), \tag{40}$$

$$K^*(k) = -[I \ 0]K_0(k) \begin{bmatrix} Z_{ux}(k) \\ \hat{C}_x(k) \end{bmatrix}.$$

If $rank(\hat{C}_x(0)) \neq 0$, then either $\hat{C}_x(0)\delta x(0) \equiv 0$, which means that the linear equations resulted from linearizing active constraints are redundant, or the variation $\delta x(0)$ for the initial state $x(0)$ is infeasible. In both cases, further modification will be needed in order to apply the proposed algorithm, which will be addressed in the next section.

6. DEALING WITH DEGENERACY USING LINEAR PROGRAMMING SOLUTION

Let us revisit the same ship control example. At the time instant $i = 74$, see point B highlighted in Figure 1, singularity arises again with the matrices $C_u^a(\cdot)$ and $\bar{C}_x^a(\cdot)$. Applying the algorithm proposed in Section 5, the back-propagation continues until $k = 0$. The resulting matrices C_u^a, $\bar{C}_x^a$ and $\hat{C}_x$ are given in Table II (to save space, the matrices at $k = 2, 3, 4, 5, 6$ are omitted). Note that at $k = 0$, we have

$$\hat{C}_x(0) = [0.0427 \ 0.1153 \ -0.0137 \ -0.0003 \ 0.0787], \tag{41}$$

which means that the following equation,

$$\hat{C}_x(0)\delta x(0) = 0, \tag{42}$$

has to be satisfied. This leads to a constraint on the initial condition which is not a manipulated variable and this leads to either degeneracy or infeasibility of the associated Quadratic Programming problem.

In this case, the following two remarks provide more insights into the causes and suggest possible mitigating strategies.

Remark 7. For a given $\delta x(0)$, if (42) is satisfied, it implies that the linear constraints in the associated QP problem are dependent. This is referred to as the "degeneracy" in the optimization literature (Gill et. al, (1981)). In this case, if these dependent constraints can be properly identified and eliminated from the outset, then the algorithm described in Section 5 can be applied and $rank(\hat{C}_x(0)) = 0$ will be satisfied for the reformulated QP problem with independent linear constraints.

Remark 8. If $\hat{C}_x(0)\delta x(0) \neq 0$, this implies that the QP problem with linear constraints is "infeasible". However, this does not necessarily means that the original optimization problem (1) with constraints (2)-(4) is infeasible. The infeasibility could be due to linearization or the threshold used in determining the constraint violation.

In this section, we introduce a linear programming formulation to deal with the degeneracy or infeasibility of quadratic programming problem.

Instead of solving the original QP problem, we now shift our attention to finding a feasible descent direction.

In order to enforce the original nonlinear constraints, we modify the linear equality constraints

$$\begin{aligned} &C_x^a(k)\delta x(k) + C_u^a(k)\delta u(k) = -C^a(x(k), u(k)), \\ &\bar{C}_x^a(k)\delta x(k) = -\bar{C}^a(x(k)), \end{aligned}$$

into the following linear inequality constraints:

$$\begin{aligned} &C_x^a(k)\delta x(k) + C_u^a(k)\delta u(k) \leq -C^a(x(k), u(k)), \\ &\bar{C}_x^a(k)\delta x(k) \leq -\bar{C}^a(x(k)). \end{aligned}$$

We now introduce the following linear programming (LP) problem:

$$\min_{\delta u(\cdot), \delta x(\cdot)} \sum_{k=0}^{N-1} (L_x(k)\delta x(k) + L_u(k)\delta u(k)) + \Phi_x(N)\delta x(N)$$

subject to:

$$\begin{aligned} &\delta x(k+1) = f_x(k)\delta x(k) + f_u(k)\delta u(k), \\ &\delta x(0) = \delta x_0, \\ &C_x(k)\delta x(k) + C_u(k)\delta u(k) \leq -C(x(k), u(k)), \\ &\bar{C}_x(k)\delta x(k) \leq -\bar{C}(x(k)). \end{aligned} \tag{43}$$

to find the next feasible direction. In solving the LP problem (43) we achieve: (i) the value of the active constraints will not be increased, but can be decreased

Table II: The content of matrices $C_u^a(\cdot)$, $\bar{C}_x^a(\cdot)$ and $\hat{C}_x(\cdot)$ for the degeneracy case

k	0	1	···	8	9	10
$C_u^a(\cdot)$	$\begin{bmatrix}0 & 0.1\end{bmatrix}$	$\begin{bmatrix}0 & 0.1\end{bmatrix}$	···	$\begin{bmatrix}-0.61 & 0\\ 0 & 0.1\end{bmatrix}$	$\begin{bmatrix}0 & -0.1\end{bmatrix}$	
$\bar{C}_x^a(\cdot)$	$\emptyset$	$\emptyset$	···	$\begin{bmatrix}-0.0017\\ -0.2792\\ 0\\ 0\\ 0\end{bmatrix}$	$\begin{bmatrix}0.0246\\ -0.2775\\ 0\\ 0\\ 0\end{bmatrix}$	$\begin{bmatrix}0.0509\\ -0.2743\\ 0\\ 0\\ 0\end{bmatrix}$
$\hat{C}_x(\cdot)^T$	$\begin{bmatrix}-0.0427\\ 0.1153\\ -0.0137\\ -0.0003\\ 0.0787\end{bmatrix}$	$\begin{bmatrix}-0.0158 & -0.0393\\ -0.0856 & -0.2136\\ 0.0120 & 0.0300\\ 0.0007 & 0.0016\\ 0.0043 & 0.0110\end{bmatrix}$	···	$\begin{bmatrix}0.0509 & 0.0246 & -0.0017\\ -0.2743 & -0.2775 & -0.2792\\ 0.0147 & 0.0073 & 0\\ -0.0012 & -0.0010 & 0\\ -0.0042 & 0.0002 & 0\end{bmatrix}$	$\begin{bmatrix}0.05 & 0.02\\ -0.27 & -0.27\\ 0.0074 & 0\\ -0.0010 & 0\\ -0.0019 & 0\end{bmatrix}$	$\begin{bmatrix}0.05\\ -0.27\\ 0\\ 0\\ 0\end{bmatrix}$

which will lead to a reduced number of active constraints for the next iteration; (ii) the original cost function will be decreased whenever possible.

Note that solving the LP problem (43) can be performed in a time efficient manner so that it will not be a barrier for fast optimization. In addition, the LP problem is solved only when the QP problem is degenerate or infeasible, identified by the condition $rank(\hat{C}_x(0)) \neq 0$.

The LP problem is solved for ship maneuvering example at $i = 74$. Note that $C^a(x,u)$ and $\bar{C}(x)$ are defined in (15) and

$$C^a(x,u) = 0$$
$$[\bar{C}^a(7)\ \bar{C}^a(8)\ \bar{C}^a(9)\ \bar{C}^a(10)] = [1.3\ 20\ 40\ 30] \times 10^{-5} \tag{44}$$

The LP problem (43) for the ship control problem can be formulated using C_u^a and C_x^a from Table II and noting that $C_x^a = 0$. The solution to the LP problem gives,

$$\begin{bmatrix}\delta u_r\\ \delta u_t\end{bmatrix} = \begin{bmatrix}-0.0844 & 0\\ -0.8439 & 0\\ -1.2167 & 0\\ -1.2197 & 0\\ -1.215 & 0\\ -1.2182 & 0\\ -0.371 & 0\\ 0.0399 & 0\\ 0.335 & 0\\ -0.0011 & 0.0131\end{bmatrix}^T . \tag{45}$$

Once this solution is applied, the activity status of constraint on u_r and u_t at $k = 4$ and $k = 10$, is respectively changed from active to inactive. Therefore, for the next iteration of neighboring extremal method, the degeneracy does not occur because of the reduced number of active constraints.

7. CONCLUSION

In this paper, a generalized neighboring extremal solution method is proposed to deal with mixed state and input constraints in optimal control of discrete-time systems. Two novel approaches were developed: one relies on the back-propagation of constraints to avoid singularity in control calculation, another uses the linear programming to address degeneracy and infeasibility of the associated QP problem when the original optimization problem has a feasible solution. The algorithm is elucidated through the ship maneuvering example, which demonstrates that the obstacle avoidance can be achieved using the proposed algorithm.

REFERENCES

T. Binder, L. Blank, H. G. Bock, R. Bulirsch, W. Dahmen, M. Diehl, T. Kronseder, W. Marquardt, J. P. Schlöder, O. von Stryk, 2001 "Introduction to model based optimization of chemical processes on moving horizons," In M. Grötschel, S. O. Krumke, J. Rambau (editors):, *Online Optimization of Large Scale Systems,* Springer-Verlag, Berlin.

S. Qin, T. Badgwill, 1997," An overview of industrial model predictive control technology," *International Conference on Chemical Process Control.*

T.A. Johansen, I. Petersen, O. Slupphaug, 2002 "Explicit sub-optimal linear quadratic regulation with state and input constraints," *Automatica*, Vol. 38, Issue 7, pp. 1099-1111.

A. Bemporad, M. Morari, V. Dua, E.N. Pistikopoulos, 2002,"The explicit linear quadratic regulator for constrained Systems," *Automatica*, Vol. 38, pp. 3-20.

P. Dufour, Y. Toure, D. Blanc, P. Laurent, 2003 "On nonlinear distributed parameter model predictive control strategy: on-line calculation time reduction and application to an experimental drying process," *Computers and Chemical Engineering*, 27, pp. 1533-1542.

R. Ghaemi, J. Sun, I. V. Kolmanovsky, 2007a "Model predictive control for constrained discrete time systems: An optimal perturbation analysis approach," *Proceedings of the American Control Conference*, New York City, pp. 3757-3762.

R. Ghaemi, J. Sun, I. Kolmanovsky, 2007b "Model predictive control for constrained discrete time systems: An integrated perturbation analysis and sequential quadratic programming approach," *Proceedings of the 46th IEEE Conference on Decision and Control*, New Orleans.

M. H. Casado, A. Fernandez, and J. Iglesis, 2001"Optimization of the course in the ship's movement by input-output linearization", *Proceedings of IFAC Conference on Control Application and Marine Systems*, pp. 419-424.

P. E. Gill, W. Murray, M. H. Wright, 1981 *Practical optimization*, Academic Press, New York.