Decentralized Formation Control including Collision Avoidance

Nopthawat Kitudomrat

Fujita Lab
Dept. of Mechanical and Control Engineering
Tokyo Institute of Technology

FL07-16-1
Seminar 01/10/2007

Today's Topics

- Introduction
- Problem Setting
- Set Invariance Theory
- Reference Governor
- Conclusion
- Future work
- Reference

Introduction

- Decentralized Control
  - Group of dynamically decoupled subsystem
  - Perform a cooperative task such as flocking and formation control
  - Recognize only neighbors
- Set Invariance Theory
  - Steer a system to target without violating constraints
- Reference Governor
  - Feasibility on stable and constrained system
  - Need to solve optimization problem

Problem Setting (1)

- Consider N_v decoupled dynamical system
- Every agent has the same model

Discrete-time time-invariant state equation

$$x^i_{k+1} = Ax^i_k + Bu^i_k \tag{1}$$

$$y^i_k = Cx^i_k = \begin{bmatrix} I & 0 \end{bmatrix} x^i_k = p^i_k, \tag{2}$$

where $k \in \mathbb{Z}_+ := \{0, 1, 2, \cdots\}$, $i \in \{1, \cdots, N_v\}$, $u^i_k \in \mathbb{R}^{n_p}$ is the control input, $x^i_k = [{p^i_k}'{v^i_k}'x_c']' \in \mathbb{R}^n (n = n_p + n_v + n_c)$ is the state vector composed of the position, velocity and state of controller.

Problem Setting (2)

Assumption 1

a desired tracking performance is achieved in the absence of the constraints with $C(I-A)^{-1}B = I$.

Assumption 2

The state x_k^i of the system (1) can be measured exactly.

Problem Setting (3)

- The interconnection of the above systems is described by

$$\mathcal{G}_k = \{\mathcal{V}, \mathcal{A}_k\} \tag{3a}$$

$$A_k(i,j) = \begin{cases} 1, & \text{if } (i,j) \in \mathcal{A}_k \\ 0, & \text{otherwise} \end{cases} \tag{3b}$$

$$\mathcal{A}_k = \{(i,j) \mid \|y_k^i - y_k^j\|_\infty \le d_s\} \tag{3c}$$

- Agent tries to avoid collision when $\|y_k^i - y_k^j\|_\infty \le d$

Note 1

- d is collision avoidance range
- d_s is connection range
- $d \le d_s$

Notations

- $x_k = (x_k^1 \ \cdots \ x_k^{N_v})$
- $x^i(t; x_0^i, r^i), \ t \in \mathbb{Z}^+$ is the evolution of the state from the initial state x_0^i which r^i is a step reference.
- $x(t; x_0, r) = (x^1(t; x_0^1, r^1), \cdots, x^{N_v}(t; x_0^{N_v}, r^{N_v}))$
- $r = [r^1 \ \cdots \ r^{N_v}]$
- $\mathcal{R} = \{r | \ \|r^i - r^j\|_\infty \geq d + \epsilon\}, \ \epsilon > 0.$ is a reference set.

Maximal output admissible set

Definition 1

The maximal output admissible set S_x is defined by

$$S_x^i = \{(x_0^i, r^i) \mid x^i(t; x_0^i, r^i) \in \mathbb{X}, r^i \in \mathbb{U}, \forall t\} \quad (4)$$

$$S_x = \{(x_0, r) \mid r \in \mathcal{R}, (x_0^i, r^i) \in S_x^i, \forall i\} \quad (5)$$

Collision Free set and Collision set

Definition 2

The collision free set $\mathbb{Y}$ is defined by

$$\mathbb{Y} = \{x|\ (x^i, x^j) \in \mathbb{Y}^{i,j} \quad \forall i, j\ (i \neq j)\} \tag{6}$$

$$\mathbb{Y}^{i,j} = \{(x^i, x^j) \mid \|y^i - y^j\|_\infty > d\} \tag{7}$$

Definition 3

The collision set $\tilde{\mathbb{Y}}$ is defined by

$$\tilde{\mathbb{Y}} = \{x|\exists i \text{ and } j(i \neq j) \text{ s.t. } (x^i, x^j) \in \tilde{\mathbb{Y}}^{i,j}\} \tag{8}$$

$$\tilde{\mathbb{Y}}^{i,j} = \{(x^i, x^j) \in \mathbb{X} \mid \|y^i - y^j\|_\infty \leq d\} \tag{9}$$

Safe Region

Definition 4

The safe region S is defined by

$$S = \{(x_0, r) \in S_x \mid (x_0^i, x_0^j, r^i, r^j) \in S^{i,j} \quad \forall i, j (i \neq j)\},$$

$$S^{i,j} = \{(x_0^i, x_0^j, r^i, r^j) \mid (x^i(t; x_0^i, r^i), x^j(t; x_0^j, r^j)) \in \mathbb{Y}^{i,j} \forall t \in \mathbb{Z}^+\}.$$

Predictive Collision Set

Definition 5

The predictive collision set is defined by

$$\begin{aligned}\tilde{\mathcal{O}}_0^{i,j} &= \{(x^i, x^j, r^i, r^j) \mid (x^i, x^j) \in \tilde{\mathbb{Y}}^{i,j}, (x^i, r^i) \in S_x^i \\ &\quad , (x^j, r^j) \in S_x^j\} \\ \tilde{\mathcal{O}}_{k+1}^{i,j} &= \{(x^i, x^j, r^i, r^j) \mid (x^i, x^j, r^i, r^j) \in \tilde{\mathcal{O}}_k^{i,j} \\ &\quad , (x^i, r^i) \in S_x^i, (x^j, r^j) \in S_x^j\} \\ \tilde{\mathcal{O}}_\infty^{i,j} &= \bigcup_{k=0}^{\infty} \tilde{\mathcal{O}}_k^{i,j} \end{aligned} \tag{10}$$

Safe Region Calculation Procedure

The algorithm for calculating the safe region S is described as:

1. Start with $k = 0$, and compute the initial predictive collision set $\tilde{\mathcal{O}}_0^{i,j}$.
2. Find $\tilde{\mathcal{O}}_{k+1}^{i,j}$ which satisfies the constraint $Ax + Br \in \tilde{\mathcal{O}}_k^{i,j}$.
3. Check if $\tilde{\mathcal{O}}_{k+1}^{i,j} \subseteq \tilde{\mathcal{O}}_k^{i,j}$
   - Yes : Go to step 4.
   - No : Increase a time step k by 1 and go to step 1.
4. Compute the safe region S by using the following equations

$$\tilde{\mathcal{O}}_\infty^{i,j} = \bigcup_{p \in \{0,\ldots,k\}} \tilde{\mathcal{O}}_p^{i,j}$$

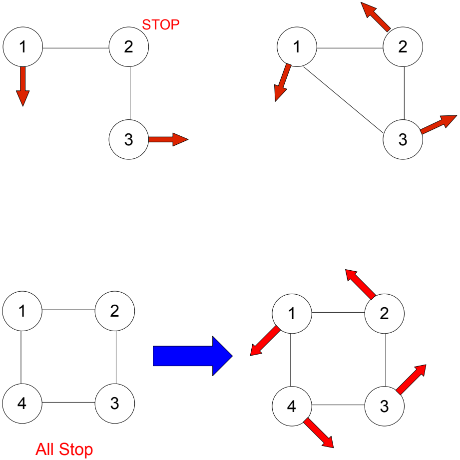
$$S^{i,j} = S_x \setminus \tilde{\mathcal{O}}_\infty^{i,j}$$

Decision Making Algorithm

The algorithm can be described in details as follow:

1. Compute the adjacency matrix A_k according to (3c)
2. Select the agent and consider its link structure if there are any neighbors or not :
   - Yes : Check the existence of a neighbor which does not connect to other neighbors.
     - Yes : The agent tries to stop and wait for a next operation. (condition 1)
     - No : Calculate the direction of movement by sum of the vectors from neighbors to the considered agent and rotate it 90 degrees. Then, we compute the input according to the algorithm shown in the next section. (condition 2)
   - No : Apply the controller that is described in the next section. (condition 3)
3. If the agent and all of its neighbors stop, then, at the next sampling time, the agent sends the information of direction of movement to all the neighbors and moves.

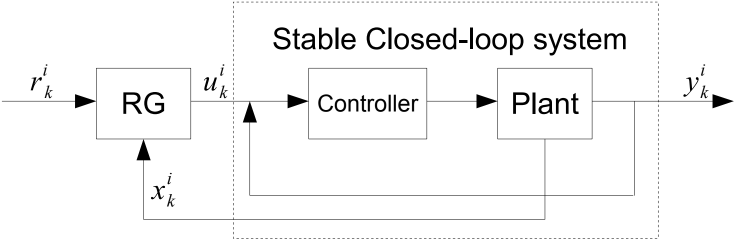
Examples



Reference governor (RG)

Properties of RG

- Deal with tracking control problem
- Modify input to satisfy the state constraints
- Consist of optimization problem



Input Computation

- We introduce a constant $K_k^i \in [0,1]$ such that $u_k^i = K_k^i r_k^i + (1 - K_k^i)\hat{u}_k^i$, where K_k^i is computed at each time step and $\hat{u}_k^i$ is an input calculated from decision making algorithm.
- Find the optimized value K^* which satisfies the collision avoidance

$$(x_k^i, x_k^j, u_k^i(K_k^i), u_k^{i,j}(K_k^j)) \in S^{i,j} \quad \forall j \quad \text{s.t. } (i,j) \in \mathcal{A}_k. \quad (11)$$

where $u_k^i(K) = Kr_k^i + (1 - K)\hat{u}_k^i$, $u_k^{i,j}(K) = K\hat{r}_k^{i,j} + (1 - K)\hat{u}_k^j$, and $\hat{r}_k^{i,j}$ is the estimate value of r_k^j from the agent i.

Input Computation (2)

The input r_k^i and $\hat{u}_k^i$ can be acquired by using the decision making algorithm.

Condition	r_k^i	$\hat{u}_k^i$
1	p_k^i	p_k^i
2	direction (magnitude ??)	p_k^i
3	terminal point	u_{k-1}^i

Input Computation (3)

- The agent does not know the next state behavior (Move, Stop) of the neighbor so both cases are considered

$$(x_k^i, u_k^i(K_k^i)) \in S_x^i \tag{12a}$$
$$(x_k^j, u_k^{i,j}(K_k^j)) \in S_x^j \quad \forall j \text{ s.t. } (i,j) \in \mathcal{A}_k \tag{12b}$$
$$(Ax_k^i + Bu_k^i(K_k^i), CAx_k^i + CBu_k^i(K_k^i)) \in S_x^i \tag{12c}$$
$$(Ax_k^j + Bu_k^{i,j}(K_k^j), CAx_k^j + CBu_k^{i,j}(K_k^j)) \in S_x^j \quad \forall j \text{ s.t. } (i,j) \in \mathcal{A}_k \tag{12d}$$
$$(x_k^i, x_k^j, u_k^i(K_k^i), \hat{u}_k^j) \in S^{i,j} \quad \forall j \text{ s.t. } (i,j) \in \mathcal{A}_k \tag{12e}$$
$$(x_k^i, x_k^j, \hat{u}_k^i, u_k^{i,j}(K_k^j)) \in S^{i,j} \quad \forall j \text{ s.t. } (i,j) \in \mathcal{A}_k \tag{12f}$$

Input Computation (4)

- The optimization problem can be formulated as follow :

$$\max_{K_k^i, K_k^j} \quad K_k^i + \sum_{(i,j)\in\mathcal{A}_k} K_k^j$$
$$\text{s.t.} \quad : K_k^i, K_k^j \in [0,1], \text{constraints(11) and (12)}$$

- In condition 3, we solve the following problem.

$$\max_{K_k^i, K_k^j} \quad K_k^i$$
$$\text{s.t.} \quad : K_k^i, K_k^j \in [0,1], \text{constraints(11) and (12a), (12c)}$$

Conclusion

We have proposed an algorithm to calculate a reference input for the stable closed-loop system. Set region which satisfies collision avoidance constraints is realized by set invariance theorem. We applied the reference governor controller to achieve the tracking performance without violating constraints.

Future Works

- Consider the minimum range of sensor (d_s)
- Calculate the magnitude of input r in condition 2 to guarantee stability.
- Simplify the calculation of K_k^i, K_k^j

References

S.V. Rakovic and D.Q. Mayne, "Robust model predictive control for obstacle avoidance: discrete time case," *Proceedings of the International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control NMPC05*, Freudenstadt - Lauterbad, Germany, August 2005, pp. 347-356.

S.V. Rakovic and D.Q. Mayne, "Robust Time Optimal Obstacle Avoidance Problem for Constrained Discrete Time Systems," *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference,* December 2005, pp. 981-986.

F. Blanchini, "Set invariance in control," *Automatica*, Vol. 35, No. 11, November 1999, pp. 1747-1767.

References (2)

- E.C. Kerrigan, "Robust Constraint Satisfaction: Invariant Sets and Predictive Control," *Ph.D. dissertation*, Control Group, Department of Engineering, University of Cambridge, England, November 2000.
- K. Hirata and M. Fujita, "Set of Admissible Reference Signals and Control of Systems with State and Control Constraints," *Proceedings of the 38th IEEE Conference on Decision and Control*, Phoenix, Arizona, USA, December 1999, pp.1427-1432.
- A. Bemporad, A. Casavola and E. Mosca, "A predictive reference governor for constrained control systems," *Computers in Industry*, Vol.36, Issue 1-2, April 1998, pp. 55-64.

Appendix Pic