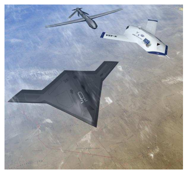
Formation Control via Receding Horizon Control: A Set Theoretic Approach

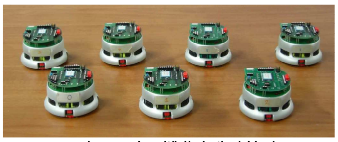
FL-08-05-1
Nopthawat Kitudomrat
19/05/2008

Introduction

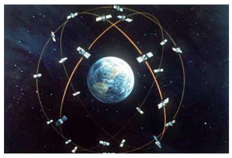
- Multi-robot System
- Formation Control
- Collision Avoidance
- Application
  - UAVs, UGVs, UUVs
  - Satellite Orbit
  - Mobile Robots



http://www.public.iastate.edu



webuser.unicas.it/lai/robotica/video/



http://www.gpsmagazine.com/

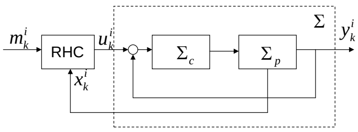
Recent works

- W.B. Dunbar, et al. (Automatica,2006)
  - Multi-robot system without collision avoidance
  - Each robot are not deviated too far from the previous open loop trajectories.
  - Receding horizon update is sufficiently fast.
- T. Keviczky, et al. (ACC,2006)
  - Multi-robot system with collision avoidance
  - Decentralized Receding Horizon Control
  - Applied coordination penalized values and using the mixed integer linear programming (MILP)
- T. Keviczky, et al.(CDC,2004)(IEEE Trans.,2008)
  - Multi-robot system with collision avoidance
  - Decentralized Receding Horizon Control
  - When feasibility is lost : Use Emergency Maneuver based on invariant set theory

Outline

- Introduction
- Problem setting
- Constraints and Conditions for constraints
- Controller Design
- Feasibility of the optimization problem
- Conclusion

Problem Setting



Consider the second ordered system with priori compensator

$$\Sigma: \quad x_{k+1}^i = Ax_k^i + Bu_k^i, \quad y_k^i = Cx_k^i = [I \quad 0]x_k^i = p_k^i$$

$$x_k^i = \begin{bmatrix} p_k^i & v_k^i \end{bmatrix}^T \in R^{n_p+n_v}, \quad i \in \{1,\dots,N_v\}, \quad k \in Z_+ := \{0,1,2,\dots\}$$

- p_k^i : Position
- v_k^i : Velocity
- m_k^i : Target Position
- u_k^i : System Input

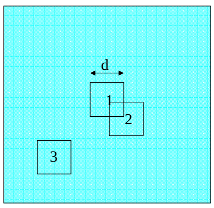
Assumption
- The matrix A is asymptotically stable
- Σ is an integral type servo system :

$$C(I-A)^{-1}B = I$$

Problem Setting

- Two agents is collision free when

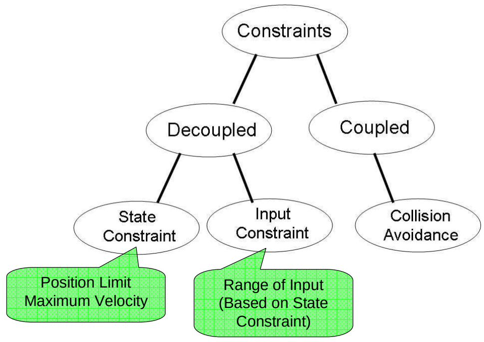
$$(x_k^i, x_k^j) \in X_{CA} := \left\{ (x^i, x^j) \,\middle|\, \left\| p^i - p^j \right\|_\infty > d \right\}$$



Outline

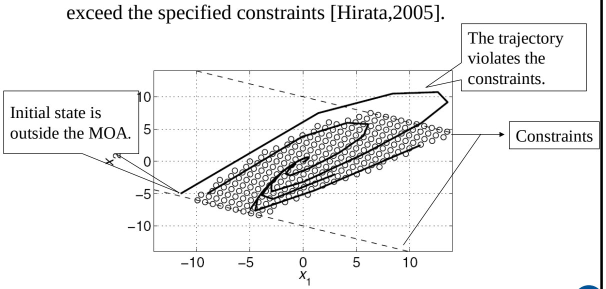
- Introduction
- Problem setting
- Constraints and Conditions for constraints
- Controller Design
- Feasibility of the optimization problem
- Conclusion

Constraints



Maximal Output Admissible Set (MOA)

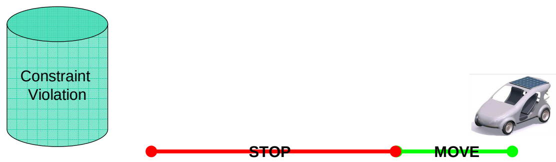
- The maximal output admissible set is the largest constraint admissible positively invariant set or, in other word, the set of all initial conditions such that the trajectories never exceed the specified constraints [Hirata,2005].



Concept for conditions in this work

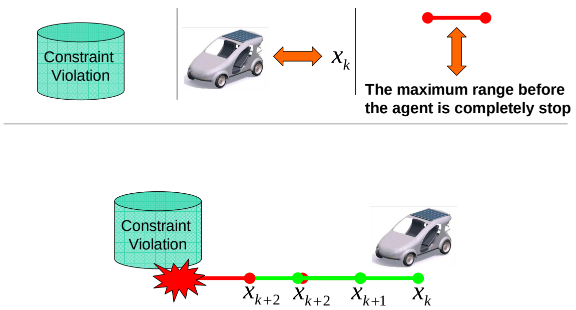
Definition :

- STOP := Move to current position (p_k^i)
  $:= (u_k^i = p_k^i)$
- Completely STOP := $(v_k^i = 0)$



Meaning : If the agent is close to violate the constraint (but still has enough space to stop), it can at least stop to avoid the violation.

Concept for conditions in this work (2)



Condition for Decoupled Constraints

- Decoupled Constraints : No effect from other agents
- From $u_k^i = p_k^i = Cx_k^i$
- The condition can be defined as

$$S_x = \{x_0 \in \mathrm{X} \mid x_k(x_0) \in \mathrm{X}, k \in Z_+\}$$

$$x_k(x_0) = (A + BC)^k x_0$$

where X is a state constraint.

Condition for Coupled Constraint

- Safe region is a set of initial conditions that guarantee the collision avoidance when the agent try to STOP.

$$S^{i,j} = \left\{ \begin{array}{l} (x_0^i, x_0^j) \mid (x_0^i, x_0^j) \in S_x \times S_x, \\ (x_k^i(x_0^i), x_k^j(x_0^j)) \in X_{CA}, k \in Z_+ \end{array} \right\}$$

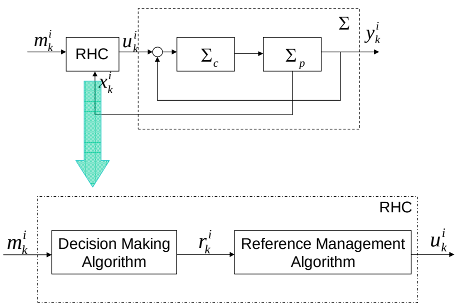
Decoupled Constraint is included !

Collision Avoidance Constraint for all trajectories

Outline

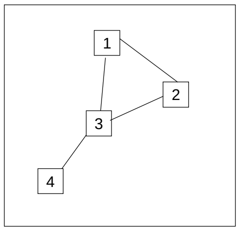
- Introduction
- Problem setting
- Constraints and Conditions for constraints
- Controller Design
- Feasibility of the optimization problem
- Conclusion

Analysis of Controller



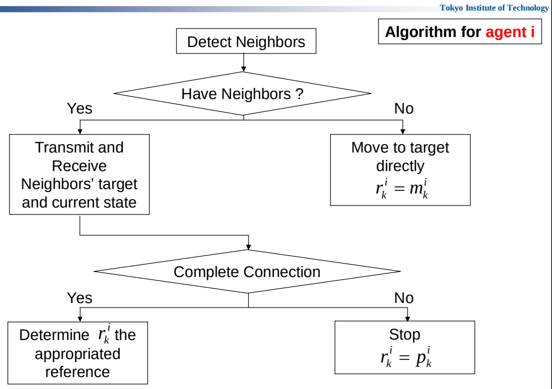
Modify the input so that the constraint fulfillment and collision avoidance are achieved.

Complete Connection



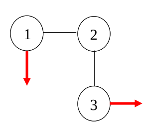
- Connection : can be detected
  $\left\| p_k^i - p_k^j \right\|_\infty \le d_s$
- Complete Connection : All neighbors can detect all of the rest.
- Agent 1,2 : Complete Connection
- Agent 3 : NOT Complete Connection
- But Agent 4 : Complete Connection

Decision Making Algorithm

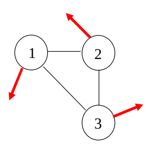


Decision Making Algorithm (2)

Example of reference selection

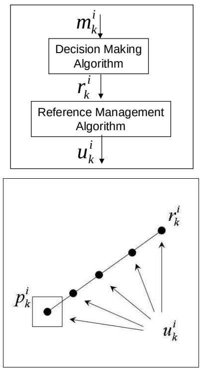


Only Agent 2 's information is not complete connection so its state is STOP



All agents are complete connection

Reference Management Algorithm



- The input is defined as follow

$$u_k^i = K_k^i p_k^i + (1-K_k^i) r_k^i,\ K_k^i \in [0,1]$$

To find the input u_k^i which is closest to the reference r_k^i

$$\min K_k^i + \sum_{j \in N_i(k)} \hat{K}_k^j$$

s.t. Model of system
Decoupled Constraint
Coupled Constraint

$N_i(k)$: Neighbors of agent i at time k

Reference Management Algorithm (2)

$$\min K_k^i + \sum_{j \in N_i(k)} \hat{K}_k^i$$

s.t. $K_k^i, \hat{K}_k^j \in [0,1]$

$u_k^i = K_k^i p_k^i + (1-K_k^i) r_k^i$, $\hat{u}_k^j = \hat{K}_k^j p_k^j + (1-\hat{K}_k^j)\hat{r}_k^j$ } Input calculation

$x_{k+1}^i = Ax_k^i + Bu_k^i$, $\hat{x}_{k+1}^j = Ax_k^j + B\hat{u}_k^j$ } State Update : Move

$\tilde{x}_{k+1}^i = Ax_k^i + Bp_k^i$, $\tilde{x}_{k+1}^j = Ax_k^j + Bp_k^j$ } State Update : Stop

$(x_{k+1}^i, \hat{x}_{k+1}^j) \in S^{i,j}$ → (Move, Move)

$(x_{k+1}^i, \tilde{x}_{k+1}^j) \in S^{i,j}$ → (Move, Stop)

$(\tilde{x}_{k+1}^i, \hat{x}_{k+1}^j) \in S^{i,j}$ → (Stop, Move)

Outline

- Introduction
- Problem setting
- Constraints and Conditions for constraints
- Controller Design
- Feasibility of the optimization problem
- Conclusion

Feasibility of the Optimization Problem

- Theorem

Suppose that the initial state $(x_0^i, x_0^j) \in S^{i,j}$ and the detection range d_s is wide enough. Then, the present receding horizon control law achieves both the constraint fulfillment and collision avoidance for any time instant

- Proof

–Consider the feasibility of the optimization problem

$(x_0^i, x_0^j) \in S^{i,j}$ ⇒ At time k+1, there is at least one input that satisfies all of the constraints which is

$u_k^i = p_k^i, \hat{u}_k^j = p_k^j$ (STOP)

⇓

Feasibility ⇒ Constraint Fulfillment, Collision Avoidance

Outline

- Introduction
- Problem setting
- Constraints and Conditions for constraints
- Controller Design
- Feasibility of the optimization problem
- Simulation and Conclusion

Simulation

$$x_{k+1}^i = \begin{bmatrix} 0.87 & 0 & 0.32 & 0 \\ 0 & 0.87 & 0 & 0.32 \\ -0.46 & 0 & 0.33 & 0 \\ 0 & -0.46 & 0 & 0.33 \end{bmatrix} x_k^i + \begin{bmatrix} 0.13 & 0 \\ 0 & 0.13 \\ 0.45 & 0 \\ 0 & 0.45 \end{bmatrix} u_k^i,\quad y_k^i = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k^i$$

$d = 0.5$

$d_s = 2.45$

$-10 \le p_k^i \le 10, -1 \le v_k^i \le 1$

$x_0^1 = [0 \quad 0 \quad 0 \quad 0]^T$

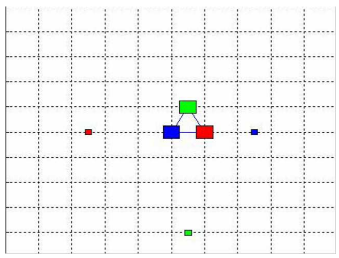
$x_0^2 = [1 \quad 0 \quad 0 \quad 0]^T$

$x_0^3 = [0.5 \quad 1 \quad 0 \quad 0]^T$

$m_0^1 = [2.5 \quad 0]^T$

$m_0^2 = [-2.5 \quad 0]^T$

$m_0^3 = [0.5 \quad -4]^T$



Conclusion and Future works

- Conclusion
  - Concept of condition formulation
  - Proposed Algorithms
  - Feasibility and collision avoidance proof
- Future works
  - Stability of the system
  - Observer-based velocity
  - Nonlinear system

THANK YOU