Frontiers in Networked Control	Lecture 3
Royal Institute of Technology (KTH)	19 April 2006

Decentralized Receding Horizon Control

Lecturer: Jens Pettersson — Scribe: David A. Anisi

The outline of todays lecture is as follows.

1. Brief introduction to RHC.
2. Motivation for a decentralized approach (DRHC).
3. Review of paper [1].
4. Review of paper [2].

1 RHC Introduction

Receding horizon control (RHC), or Model Predictive Control (MPC), is an optimization based control method mainly adopted in the process industry. It is based on the online solution of the following Dynamic Optimization Problem (DOP)

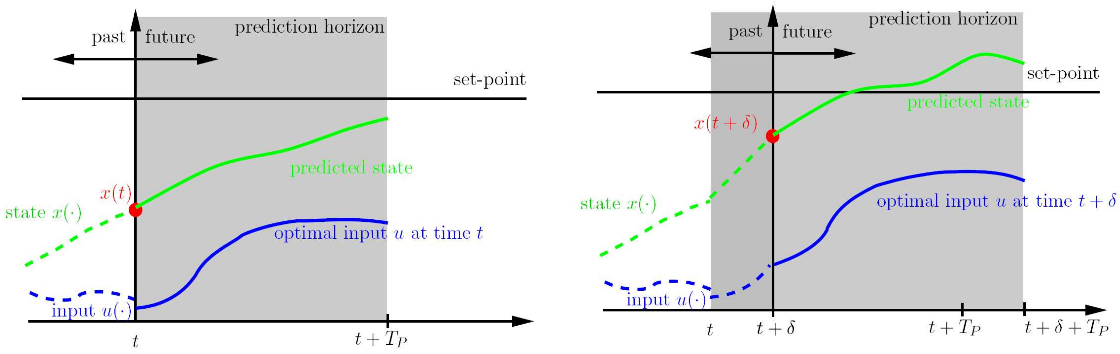
$$\begin{aligned} \underset{u,p}{\text{minimize}} \quad & \int_{t_c}^{t_c+T_p} \mathcal{L}(x(t), u(t))\mathrm{d}t + \Psi(x(t_c+T_p)) \\ \text{subject to} \quad & F(\dot{x}(t), x(t), u(t), p, w(t)) = 0 \\ & L \leq \begin{bmatrix} x(t) \\ u(t) \\ \dot{u}(t) \end{bmatrix} \leq U \end{aligned} \tag{1}$$

Nomenclature:

x	State vector.
u	Control vector.
p	Parameter vector.
w	Disturbance vector.
t_c	"Current" or "considered" time instance.
T_p	Planning horizon.
L, U	Lower and upper bounds.

The RHC algorithm:

1. Measure (or estimate) the current state, $x(t_c)$.
2. Solve DOP (1) with $x(t_c)$ as the initial condition.
3. Implement $u^*(\tau)$ for $\tau \in [t_c, t_c + \delta]$.
4. Move the time horizon forward by setting $t_c = t_c + \delta$.
5. Goto 1.



(a) Current time: $t_c = t$.

(b) Current time: $t_c = t + \delta$.

Figure 1: Schematic representation of the RHC scheme.

A graphical representation of the principal of RHC can be seen in Figure 1.

History of RHC:

Year	Product	Model
1973	DMC (Shell)	Unconstrained linear models
1974	IDCOM	Constrained linear models

RHC pros:

- Handles constraints.
- Handles non-square systems (*e.g.* fat or slim systems).
- Easy to tune and maintain (bumb-tests).
- Easy to implement (purchase software tools).
- Very good track record.

RHC cons:

- Requires very heavy computations.
- Non-predictable sampling time.
- Unstable process.
- Robustness.

To conclude: “RHC is the future of automatic control”!

2 Decentralized Receding Horizon Control (DRHC)

Consider a set of subsystems that influence each other, either through the cost function, constraints and/or dynamic coupling. There is a major increase in the computational effort for a centralized RHC scheme of the overall system. To keep the run-time at check, DRHC assigns a local RHC to each subsystem. Non-interacting subsystem might then solve their local DOP in parallel. Notice that the principal argument for adopting DRHC is *not* a restricted communication bandwidth.

DRHC implementation:

- Set up a local RHC for each subsystem.
- Add the influence from all other subsystems.
- Add additional constraints.
- Solve the local DOP.

3 Review of Paper [1]

This paper considers a distributed LTI discrete time system

$$x(k+1) = Ax(k) + Bu(k) + Ew(k), \tag{2}$$

with

$$\begin{aligned} \text{state} \quad & x(k) = [x_1(k), \cdots x_M(k)]^T, \\ \text{control} \quad & u(k) = [u_1(k), \cdots u_M(k)]^T, \\ \text{disturbance} \quad & w(k) = [w_1(k), \cdots w_M(k)]^T. \end{aligned}$$

Each of the M (linear) subsystems is described by the state vector $x_i \in \mathbb{R}^{n_i}$ and the control vector $u_i \in R^{m_i}$ and is further assumed to be controllable with only state variable couplings. This can be modeled as

$$\begin{aligned} A &= \begin{bmatrix} A_{11} & \cdots & A_{1M} \\ \vdots & \ddots & \vdots \\ A_{M1} & \cdots & A_{MM} \end{bmatrix} \\ B &= \begin{bmatrix} B_1 & & \\ & \ddots & \\ & & B_M \end{bmatrix} \\ E &= \begin{bmatrix} E_1 & & \\ & \ddots & \\ & & E_M \end{bmatrix} \end{aligned} \tag{3}$$

where B_i is of full rank and

$$\text{rank}(B_i A_{i,1} \cdots A_{i,i-1} A_{i,i+1} \cdots A_{iM} E_i) = m_i \quad \forall i \in \{1, \cdots, M\}.$$

The two last conditions serve as a sufficient conditions for system (2) to be transformed to what the authors refer to as the “decentralized controllable companion form” which implies that the subsystems are connected only by state and not control nor disturbance.

In order to set stage for presenting the DRHC algorithm proposed in the paper, a few words on the adopted notation might be fruitful. In what follows, $\hat{\alpha}$ denotes predicted values of the corresponding variable, α. Let then

$$\begin{aligned} X_i(k) &= \{\hat{x}_i(k+1|k), \cdots, \hat{x}_i(k+N|k)\}, \\ U_i(k) &= \{\hat{u}_i(k|k), \cdots, \hat{u}_i(k+N-1|k)\}, \end{aligned}$$

denote the predicted values of the state and control respectively. Here N is the prediction horizon. The one-step delayed prediction exchange for subsystem i is written as

$$\hat{v}_i = \left[\hat{x}_1(k+j|k-1), \cdots, \hat{x}_{i-1}(k+j|k-1), \hat{x}_{i+1}(k+j|k-1), \cdots \hat{x}_M(k+j|k-1)\right].$$

The paper then proposes that the i^{th} controller is designed according to the following algorithm.

DRHC with Stability Constraint (DRHC-SC)

Step 1. Send out $X_i(k)$ and receive

$$V_i(k) = \{\hat{v}_i(k|k), \cdots, \hat{v}_i(k+N-1|k)\}$$

from other controllers.

Step 2. Measure the current state and disturbance, *i.e.* $x_i(k)$ and $w_i(k)$, but also the step-length parameter $l_i(k)$. Define

$$\hat{l}_i(k) = \max\{l_i(k), \|x_i(k)\|^2\} - \beta\|x_i^1(k)\|^2,$$

with the design-parameter $\beta \in (0, 1]$. Next, predict the future disturbances

$$\hat{w}_i(k+1|k), \cdots, \hat{w}_i(k+N-1|k).$$

Step 3.

$$\begin{aligned} \underset{U_i}{\text{minimize}} \quad & J_i(X_i, U_i) \\ \text{subject to} \quad & \hat{x}_i(k+j+1|k) = A_{ii}\hat{x}_i(k+j|k) + B_i\hat{u}_i(k+j|k) + K_i\hat{v}_i(k+j|k) + E_i\hat{w}_i(k+j|k) \\ & j = 0, 1, \cdots, N-1 \\ & \|\hat{x}_i(k+1|k)\|^2 \leq \hat{l}_i(k) \quad \text{(stability contstraint)}. \end{aligned}$$

Step 4. Let

$$\begin{aligned} u_i(k) &= \hat{u}_i(k|k) \\ l_i(k+1) &= \|\hat{x}_i(k+1|k)\|^2. \end{aligned}$$

Step 5. Implement control $u_i(k)$, set $k = k+1$ and return to **Step 1.** at the next sample time.

The following theorem reveals the main stability result of the paper.

Theorem 1 *The DRHC-SC algorithm is stable if*

$$\bar{A} = \begin{bmatrix} 0 & A_{12}^2 & \cdots & A_{1M}^2 \\ A_{21}^2 & 0 & \cdots & A_{2M}^2 \\ \vdots & \vdots & \ddots & \vdots \\ A_{M1}^2 & A_{M2}^2 & \cdots & 0 \end{bmatrix}$$

is stable, i.e. *has all eigenvalues inside the unit circle.*

4 Review of Paper [2]

This papers considers the distributed control of interacting subsystems whose dynamics and constraints are non-interacting, but who are coupled through a non-separable cost function. The main objective is to stabilize the formation of a group of N vehicles toward an equilibrium point in a cooperated way using RHC. The dynamics of the vehicles in the group are modeled as

$$\dot{z}_i(t) = f_i(z_i(t), u_i(t)), \quad i \in \{1, \cdots, N\}$$

so that the dynamics can be different for each vehicle. Here, the state vector, $z_i = (q_i, \dot{q}_i) \in \mathbb{R}^{2n}$ is a position-velocity pair, while the control vector u_i is restricted to a compact subset in $\mathbb{R}^m$, denoted $\mathcal{U}$. Regarding the vector fields, $f_i,\ i \in \{1, \cdots, N, \}$, we assume them to be all stabilizable at an equilibrium point, $(z_c, 0)$. Given three design-parameters, $w, \nu, \mu \in \mathbb{R}$, the centralized objective function for multi-vehicle formation stabilization is

$$\mathcal{L}(z, u) = \sum_{(i,j)\in\mathcal{E}_0} w\|q_i - q_j + d_{ij}\|^2 + w\|q_\Sigma - q_d\|^2 + \nu\|\dot{q}\|^2 + \mu\|u\|^2, \tag{4}$$

where $\mathcal{E}_0$ is the set of all pair-wise neighbors such that

- For all $i \in \{1, \cdots, N, \},\ (i, i) \notin \mathcal{E}_0$.
- Every vehicle has at least one neighbor, *i.e.* for all $i \in \{1, \cdots, N, \},\ \exists j \neq i : (i, j) \in \mathcal{E}_0$ or $(j, i) \in \mathcal{E}_0$.
- $(i, j) \in \mathcal{E}_0 \Rightarrow (j, i) \notin \mathcal{E}_0$.

Additionally, the given constants d_{ij} denote the desired distance between vehicle i and j in the formation. The second term in the cost function, is referred to as the *tracking cost* of the three leading vehicles and defined by $q_\Sigma = \frac{1}{3}(q_1 + q_2 + q_3)$ and $q_d = \frac{1}{3}(q_1^c + q_2^c + q_3^c)$.

Nomenclature:

z_i, u_i	State and control of vehicle i.
$\mathcal{N}_i$	Neighbors of vehicle i.
$z_{-i} = (z_{j1}, \cdots, z_{jk})$	State of the neigbors of vehicle i.
$u_{-i} = (u_{j1}, \cdots, u_{jk})$	Control of the neigbors of vehicle i.
$\dot{z}_{-i} = f_{-i}(z_{-i}, u_{-i})$	Collective dynamics of the neighbors of vehicle i.
$z_i^p(\cdot\,; t_k), u_i^p(\cdot\,; t_k)$	Predicted state and control trajectory for vehicle i.
$z_i^*(\cdot\,; t_k), u_i^*(\cdot\,; t_k)$	Optimal predicted state and control trajectory for vehicle i.
$\hat{z}_i(\cdot\,; t_k), \hat{u}_i(\cdot\,; t_k)$	Assumed state and control trajectory for vehicle i.
$\hat{z}_{-i}(\cdot\,; t_k), \hat{u}_{-i}(\cdot\,; t_k)$	Assumed state and control trajectory for the neighbors of vehicle i.

In what follows, the centralized objective function (4) is decomposed into distributed cost functions, $\mathcal{L}_i(z_i, z_{-i}, u_i),\ i \in \{1, \cdots, N, \}$. Then, the distributed optimal control problems and the corresponding DRHC algorithm is presented. This paper review is concluded with presenting the key requirements for stability of the algorithm presented.

The cost function in the distributed optimal control problem solved by each vehicle is defined as

$$\mathcal{L}_i(z_i, z_{-i}, u_i) = \sum_{j\in\mathcal{N}_i} \frac{\gamma w}{2}\|q_i - q_j + d_{ij}\|^2 - \gamma\nu\|\dot{q}_i\|^2 + \gamma\mu\|u_i\|^2 + L_d(i),$$

where

$$L_d(i) = \begin{cases} \gamma\frac{w}{3}\|q_\Sigma - q_d\|^2, & \text{if } i \in \{1, 2, 3\} \\ 0 & \text{otherwise} \end{cases}$$

and $\gamma \in \mathbb{R}$ is a positive constant. Notice that by construction we have

$$\sum_{i=1}^{N} \mathcal{L}_i(z_i, z_{-i}, u_i) = \gamma \mathcal{L}(z, u).$$

Before presenting the distributed optimal control problem that has to be solved on-line repeatedly by each vehicle, it must be mentioned that this paper adopts a so called *terminal controller*, which is a local feasible linear feedback, $u_i = K_i(z_i - z_i^c)$, which stabilizes each vehicle whenever we are in the vicinity of z_c. As in [3], the terminal controller is however never actually employed, but is merely used for the proof of stability.

Distributed Optimal Control Problem (DOCP)

For a vehicle i, and at any update time $t_k = t_0 + k\delta$: Given $z_i(t_k), z_{-i}(t_k), \hat{u}_i(\tau; t_k), \hat{u}_{-i}(\tau; t_k)$ for all $\tau \in [t_k, t_k + T_p]$, constants $\kappa, \varepsilon_i \in (0, \infty)$ and symmetric weighting matrices $P_i > 0$

$$\begin{array}{lll}
\underset{u_i^p}{\text{minimize}} & \displaystyle\int_{t_k}^{t_k+T_p} \mathcal{L}_i(z_i^p(s; t_k), \hat{z}_{-i}(s; t_k), u_i^p(s; t_k))\mathrm{d}s + \gamma \|z_i^p(t_k + T_p; t_k) - z_i^c\|_{P_i}^2 & \\
\text{subject to} & z_i^p(\tau; t_k) = f_i(z_i^p(\tau; t_k), u_i^p(\tau; t_k)) & \\
& \dot{\hat{z}}_i(\tau; t_k) = f_i(\hat{z}_i(\tau; t_k), \hat{u}_i(\tau; t_k)) & \\
& \dot{\hat{z}}_{-i}(\tau; t_k) = f_{-i}(\hat{z}_{-i}(\tau; t_k), \hat{u}_{-i}(\tau; t_k)) & \\
& u_i^p(\tau; t_k) \in \mathcal{U} & \\
& \|z_i^p(\tau; t_k) - \hat{z}_i(\tau; t_k)\| \leq \delta^2 \kappa & \text{(Consistency constraint)} \\
& z_i^p(t_k + T_p; t_k) \in \Omega_i(\varepsilon_i) & \text{(Terminal stability constraint)}
\end{array}$$

with initial conditions $z_i^p(t_k; t_k) = \hat{z}_i(t_k; t_k) = z_i(t_k)$ and $\hat{z}_{-i}(t_k, t_k) = z_{-i}(t_k)$. Here, the terminal set $\Omega_i(\varepsilon_i) = \{z \in \mathbb{R}^{2n} : \|z - z_i^c\|_{P_i}^2 \leq \varepsilon_i\}$.

By virtue of the consistency constraint, the optimized state for vehicle i is enforced to be at most a distance of $\delta^2\kappa$ away from what the other vehicles assume. The paper then proposes that the i^{th} controller is designed according to the following algorithm.

DRHC with Stability and Consistency Constraints

Step 1. Initialize, *e.g.* by solving the DOCP without the state compatibility constraint.

Step 2. Controller:

1. Apply $u_i^*(\tau; t_k), \quad \tau \in [t_k, t_{k+1}]$.
2. Compute $\hat{u}_i(\tau; t_{k+1}) = \hat{u}_i(\tau)$ as

$$\hat{u}_i(\tau) = \begin{cases} u_i^*(\tau, ; t_k), & \tau \in [t_{k+1}, t_k + T_p) \\ K_i(z_i(\tau) - z_i^c)) & \tau \in [t_k + T_p, t_{k+1} + T_p]. \end{cases}$$

Step 3. Transmit $\hat{u}_i(\cdot, t_{k+1})$ to every neighbor and receive $\hat{u}_j(\cdot, t_{k+1})$ from every neighbor $j \in \mathcal{N}_i$.

Step 4. At any time t_k:

1. Measure current state $z_i(t_k)$ and measure or receive the states of the neighbors $z_{-i}(t_k)$.
2. Solve DOCP yielding $u_i^*(\cdot; t_k)$ and redo.

The two key requirements for stability of the DRHC algorithm described here-above, are the consistency constraints (which force each subsystem not deviate too far from the previous open-loop state trajectory) and that the receding horizon updates happen sufficiently fast. More precisely, the authors show that there exist an upper update period bound

$$\delta_{\max} = \frac{\gamma(r/2)^2 \lambda_{\min}(Q)}{\xi + \gamma \mathcal{K} \rho_{\max}(\rho_{\max} + u_{\max}) \lambda_{\max}(Q)}.$$

These serve as sufficient conditions for stability of the proposed algorithm.

References

[1] Jia, D., Krogh, B. H., and Talukdar, S., "Distributed Model Predictive Control," *IEEE Control Systems Magazine*, Vol. 22, No. 1, Feb. 2002, pp. 44–52.

[2] Dunbar, W. B. and Murray, R. M., "Distributed receding horizon control for multi-vehicle formation stabilization," *Automatica J. IFAC*, Vol. 42, No. 4, 2006, pp. 549–558.

[3] Chen, H. and Allgöwer, F., "A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability," *Automatica J. IFAC*, Vol. 34, No. 10, 1998, pp. 1205–1217.