

Target Tracking with Dynamic Convex Optimization

Alec Koppel*, Andrea Simonetto[†], Aryan Mokhtari*,
Geert Leus[†], Alejandro Ribeiro*

*University of Pennsylvania, Philadelphia, PA

[†]Delft University of Technology, Delft, The Netherlands

Global Conference on Signal & Information Processing
Orlando, FL, Dec. 16, 2015

$$\mathbf{x}^*(t) := \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} f(\mathbf{x}; t), \quad \text{for } t \geq 0.$$

- ▶ Objective $f(\cdot, t) \Rightarrow$ time-varying cost
- ▶ For fixed time $t \Rightarrow$ convex program
- ▶ Examples problems:
 - \Rightarrow finding control gains
 - \Rightarrow target tracking
 - \Rightarrow statistical model parameters
- ▶ Focus: iterative algorithms

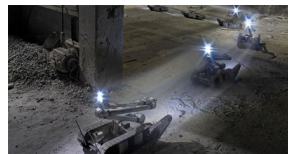
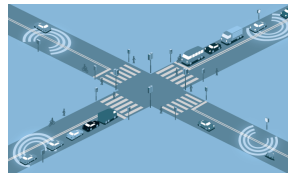
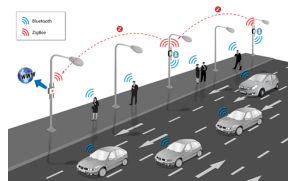


Photo credits: Libelium and iRobot

$$\mathbf{x}^*(t) := \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}; t), \quad \text{for } t \geq 0.$$

- ▶ One approach: sample at discrete times t_k

$$\mathbf{x}^*(t_k) := \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} f(\mathbf{x}; t_k).$$

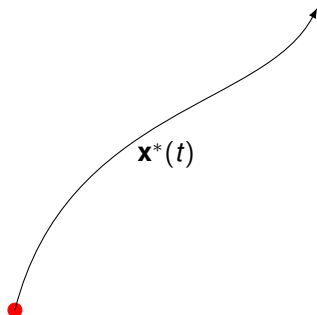
- ▶ Find $\mathbf{x}^*(t_k)$ via Newton or gradient?

⇒ only viable if $\mathbf{x}^*(t) \approx \mathbf{x}^*$

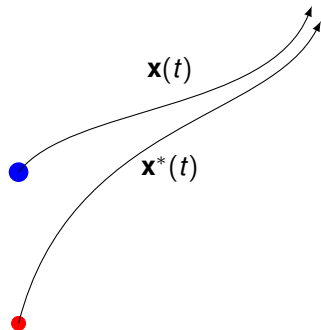
⇒ otherwise, minimum drifts:

$$\mathbf{x}^*(t_k) \neq \mathbf{x}^*(t_{k+1})$$

- ▶ Solution is an entire **trajectory**.



- ▶ Goal: $\mathbf{x}_k = \mathbf{x}(t_k)$ via $f(\mathbf{x}, t_k)$ s.t. $\mathbf{x}_k \rightarrow \mathbf{x}^*(t_k)$
 - ⇒ Main challenge: distinct $\mathbf{x}^*(t)$ for each t
 - ⇒ Stationary convex opt. methods not viable
- ▶ Instead, adapt iteration to time variation
 - ⇒ Predict $\tilde{\mathbf{x}}(t_{k+1})$ from $f(\cdot, t_k)$
 - ⇒ Correct $\tilde{\mathbf{x}}(t_{k+1})$ based on $f(\cdot, t_{k+1})$
- ▶ Main idea: **track** how $\mathbf{x}^*(t_k)$ varies in time



- ▶ Sample the problem at time instances t_k , where $h = t_k - t_{k-1}$
- ▶ We develop a method which converges to $O(h^2)$ nbhd. of $\mathbf{x}^*(t)$
 - ⇒ based on prediction-correction scheme
 - ⇒ even tighter $O(h^4)$ for Newton-based correction
- ▶ Track optimal trajectory \approx error-free

- ▶ Non-stationary opt. [Gupal, Kheisin, Nurminkii, etc., 1970s]
⇒ prediction only, require knowledge of initial optimizer $\mathbf{x}^*(t_0)$
- ▶ Parametric programming [Robinson, Rockafellar et.al., 1980s-]
⇒ prediction only; also require initial optimizer $\mathbf{x}^*(t_0)$
- ▶ Distributed signal processing & control (2010-on)
⇒ Dual decomp. [Jakubeic], ADMM [Boyd], DGD [Nedich]
⇒ Correction only, arbitrary initialization \mathbf{x}_0
- ▶ All track $\mathbf{x}^*(t)$ to $O(h)$ neighborhood; sensitive to $\mathbf{x}^*(t_0)$,

- ▶ Denote $\mathbf{x}_{k+1|k}$ as *prediction* of $\mathbf{x}^*(t_{k+1})$ using objective $f(\mathbf{x}, t_k)$
- ▶ Prediction via Newton step

$$\mathbf{x}_{k+1|k} = \mathbf{x}_k - h(\nabla_{\mathbf{xx}}f(\mathbf{x}_k; t_k))^{-1} \nabla_{\mathbf{tx}}f(\mathbf{x}_k; t_k)$$

- $\Rightarrow \nabla_{\mathbf{xx}}f(\mathbf{x}_k; t_k) \Rightarrow$ Hessian of objective at time t_k
- \Rightarrow Step-size h given as sampling rate
- \Rightarrow derived via Euler integration of optimality condition residual

- ▶ Prediction step \Rightarrow from optimality condition $\nabla_{\mathbf{x}} f(\mathbf{x}^*(t); t) = \mathbf{0}$
- ▶ For any other vector \mathbf{x} , we have $\nabla_{\mathbf{x}} f(\mathbf{x}; t) = \mathbf{r}(t)$
 $\Rightarrow \mathbf{r}(t)$ is residual vector
- ▶ Track dynamics of residual via first-order difference eqn.

$$\nabla_{\mathbf{x}} f(\mathbf{x}; t) + \nabla_{\mathbf{xx}} f(\mathbf{x}; t) \delta \mathbf{x} + \nabla_{t\mathbf{x}} f(\mathbf{x}; t) \delta t = \mathbf{r}(t),$$

- ▶ δt is variation of t . Divide above eqn. by δt , take limit $\delta t \rightarrow 0$. . .

$$\dot{\mathbf{x}} = - (\nabla_{\mathbf{xx}} f(\mathbf{x}; t))^{-1} \nabla_{t\mathbf{x}} f(\mathbf{x}; t)$$

- ▶ Using first-order Forward Euler integration, obtain prediction step

- ▶ **Prediction** $\mathbf{x}_{k+1|k}$ of $\mathbf{x}^*(t_{k+1})$ obtained from Euler integration

$$\mathbf{x}_{k+1|k} = \mathbf{x}_k - h (\nabla_{\mathbf{xx}} f(\mathbf{x}_k; t_k))^{-1} \nabla_{t\mathbf{x}} f(\mathbf{x}_k; t_k)$$

⇒ uses $f(\mathbf{x}; t_k)$ objective at time t_k

- ▶ **Correction** ⇒ either gradient or Newton step using $f(\mathbf{x}; t_{k+1})$

$$\mathbf{x}_{k+1} = \mathbf{x}_{k+1|k} - \begin{cases} \gamma \nabla_{\mathbf{x}} f(\mathbf{x}_{k+1|k}; t_{k+1}) \\ (\nabla_{\mathbf{xx}} f(\mathbf{x}_{k+1|k}; t_{k+1}))^{-1} \nabla_{\mathbf{x}} f(\mathbf{x}_{k+1|k}; t_{k+1}) \end{cases}$$

⇒ Multiple correction steps τ possible for large enough h

- ▶ Initialize \mathbf{x}_0 arbitrarily, set $h = t_{k+1} - t_k$, step-size $\gamma > 0$.
- ▶ For $k = 1, 2, \dots$

1. *Predict* via Euler-integration of sub-optimality residual

$$\mathbf{x}_{k+1|k} = \mathbf{x}_k - h(\nabla_{\mathbf{xx}}f(\mathbf{x}_k; t_k))^{-1} \nabla_{t\mathbf{x}}f(\mathbf{x}_k; t_k)$$

⇒ uses $f(\mathbf{x}; t_k)$ objective at time t_k

2. *Correct* predicted trajectory using gradient step

$$\mathbf{x}_{k+1} = \mathbf{x}_{k+1|k} - \gamma \nabla_{\mathbf{x}}f(\mathbf{x}_{k+1|k}; t_{k+1})$$

⇒ uses objective $f(\mathbf{x}; t_{k+1})$ at time t_{k+1}

- ▶ Initialize \mathbf{x}_0 arbitrarily, set $h = t_{k+1} - t_k$, step-size $\gamma > 0$.
- ▶ For $k = 1, 2, \dots$

1. *Predict* via Euler-integration of sub-optimality residual

$$\mathbf{x}_{k+1|k} = \mathbf{x}_k - h(\nabla_{\mathbf{xx}}f(\mathbf{x}_k; t_k))^{-1} \nabla_{\mathbf{tx}}f(\mathbf{x}_k; t_k)$$

⇒ uses $f(\mathbf{x}; t_k)$ objective at time t_k

2. *Correct* predicted trajectory using Newton step

$$\mathbf{x}_{k+1} = \mathbf{x}_{k+1|k} - (\nabla_{\mathbf{xx}}f(\mathbf{x}_{k+1|k}; t_{k+1}))^{-1} \nabla_{\mathbf{x}}f(\mathbf{x}_{k+1|k}; t_{k+1})$$

⇒ uses objective $f(\mathbf{x}; t_{k+1})$ at time t_{k+1}

- ▶ The function f is smooth in \mathbf{x} and uniformly in t :
 - ⇒ Twice differentiable
 - ⇒ m -strongly convex ⇒ eigenvalues of Hessian lower bounded
 - ⇒ Gradients of f are L -Lipschitz continuous
- ▶ Bounded higher-order derivatives with respect to both \mathbf{x} and t
 - ⇒ the mapping $t \mapsto \mathbf{x}^*(t)$ is one-to-one and locally Lipschitz

Theorem

i) The GTT sequence $\{\mathbf{x}_k\}$ converges to nbhd. of optimal $\mathbf{x}^*(t)$

$$\lim_{k \rightarrow \infty} \|\mathbf{x}_k - \mathbf{x}^*(t_k)\| = O(h)$$

ii) If sampling rate h and step-size γ are sufficiently small,

$$\lim_{k \rightarrow \infty} \|\mathbf{x}_k - \mathbf{x}^*(t_k)\| = O(h^2)$$

\Rightarrow Convergence rate is **Q-linear** if step-size satisfies $\gamma < 2/L$.

\Rightarrow Sampling rate threshold: $h < \left[\frac{C_0 C_1}{m^2} + \frac{C_2}{m} \right]^{-1} (\rho^{-1} - 1)$.

$\Rightarrow \rho = \max\{|1 - \gamma m|, |1 - \gamma L|\}$ \Rightarrow step-size, cvx, Lipschitz constant

Theorem

With $0 < K$ depending on grad. and cvx. constants, if rate h satisfies

$$h \leq \min \{1, K\},$$

and the algorithm is initialized such that \mathbf{x}_0 satisfies

$$\|\mathbf{x}_0 - \mathbf{x}^*(t_0)\| \leq ch^2$$

Then NTT sequence $\{\mathbf{x}_k\}$ converges to nbhd. of optimal $\mathbf{x}^*(t)$

$$\lim_{k \rightarrow \infty} \|\mathbf{x}_k - \mathbf{x}^*(t_k)\| = O(h^4)$$

- ▶ In many application domains, $\nabla_{t\mathbf{x}}f(\mathbf{x}_k; t_k)$ is not known
⇒ can be approximated by a first-order backward derivative

$$\tilde{\nabla}_{t\mathbf{x}}f(\mathbf{x}_k; t_k) \approx \frac{\nabla_{\mathbf{x}}f(\mathbf{x}_k; t_k) - \nabla_{\mathbf{x}}f(\mathbf{x}_{k-1}; t_{k-1})}{h}$$

- ⇒ situations in which exact motion of target unknown
- ▶ Can establish comparable convergence guarantees to exact algs.

Theorem

GTT method with approx. time-derivative in prediction step yields

- i) *If step-size satisfies $\gamma < 2/L$, $\{\mathbf{x}_k\} \rightarrow \mathbf{x}^*(t_k)$ Q-linearly rate to $O(h)$ bounded error*

$$\|\mathbf{x}_k - \mathbf{x}^*(t_k)\| \leq \rho^k \|\mathbf{x}_0 - \mathbf{x}^*(t_0)\| + O(h) + O(h^2)$$

- ii) *For small enough h and $\gamma > 0$, $\{\mathbf{x}_k\} \rightarrow \mathbf{x}^*(t_k)$ Q-linearly up to a bounded $O(h^2)$ error*

$$\|\mathbf{x}_k - \mathbf{x}^*(t_k)\| \leq (\rho\sigma)^k \|\mathbf{x}_0 - \mathbf{x}^*(t_0)\| + O(h^2)$$

Theorem

Suppose the NTT sampling rate satisfies

$$h \leq \min \{1, K'\},$$

where K' depends on grad. & cvx. constants, and is initialized so that

$$\|\mathbf{x}_0 - \mathbf{x}^*(t_0)\| \leq ch^2$$

Then the alg. achieves an $O(h^4)$ asymptotic tracking error

$$\|\mathbf{x}_k - \mathbf{x}^*(t_k)\| \leq O(h^4)$$

Target Tracking Example

- Scalar numerical example

$$f(x; t) := \frac{1}{2} (x - \cos(\omega t))^2 + \frac{\kappa}{2} \sin^2(\omega t) \exp(\mu x^2)$$

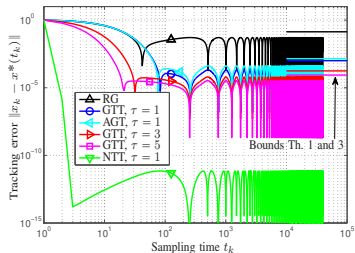
- Predictor-corrector methods track $\mathbf{x}^*(t)$
 \Rightarrow magnitudes less in tracking error

- Worst-case error:

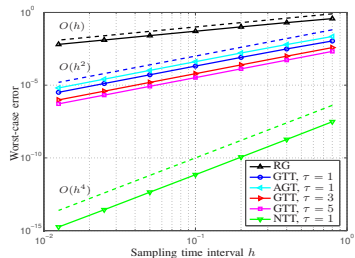
$$\max_{k > \bar{k}} \{\|x_k - x^*(t_k)\|\}$$

where $\bar{k} = 10^4$.

- \Rightarrow performance improvement clearer
- Newton more accurate but higher cost



(a) Tracked sample planar trajectory



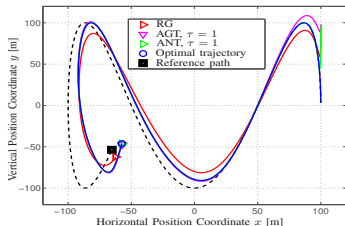
(b) Worst-case error

- ▶ Target tracking example

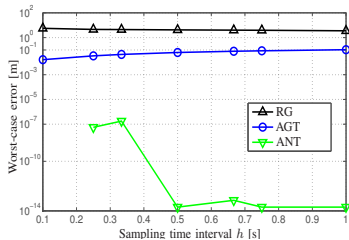
$$f(\mathbf{x}; t) := \frac{1}{2} \left(\|\mathbf{x} - \mathbf{y}(t)\|^2 + \mu_1 \exp(\mu_2 \|\mathbf{x} - \mathbf{b}\|^2) \right)$$

- ▶ Track a moving planar object $\mathbf{y}(t)$
 - ⇒ stay close to base station \mathbf{b}
- ▶ Predictor-corrector methods track optimal

- ▶ Worst-case error:
 $\max_{k > \bar{k}} \{\|x_k - x^*(t_k)\|\}$, with $\bar{k} = 10^4$
- ▶ Fix comp. cost for given sampling rate
- ▶ Newton computationally heavier
 - ⇒ higher control latency
- ▶ Main takeaway: use NTT (ANT) if sampling rate large enough



(a) Tracked sample planar trajectory



(b) Worst-case error

- ▶ Focused on continuously-varying convex opt. problems
- ▶ Brute force approach intractable \Rightarrow discrete-time sampling

- ▶ Developed prediction-correction methods
 - \Rightarrow **Predict** where optimal trajectory will be at next time
 - \Rightarrow Once we observe info. at next sample time, make **correction**
- ▶ Correction step \Rightarrow Gradient or Newton steps (GTT or NTT)

- ▶ Theorem: GTT & NTT track $\mathbf{x}^*(t)$ up to $O(h^2)$ or better
 - \Rightarrow **Correction reduces tracking error relative to existing methods**
- ▶ No problem if we don't know exact objective time variation

- ▶ Numerical examples illustrate utility in control domain
 - \Rightarrow When sampling rate is large enough, better to use NTT

- ▶ A. Simonetto, A. Koppel, A. Mokhtari, G. Leeus, and A. Ribeiro, “A Class of Prediction-Correction Methods for Time-Varying Convex Optimization,” IEEE Trans. Signal Process (submitted)., Sept. 2015.
- ▶ A. Koppel, A. Simonetto, A. Mokhtari, G. Leus, and A. Ribeiro, “Target Tracking with Dynamic Convex Optimization”, IEEE Global Conference on Signal and Information Processing, Orlando, FL, Dec. 14-16, 2015.
- ▶ A. Simonetto, A. Koppel, A. Mokhtari, G. Leeus, and A. Ribeiro “Prediction-Correction Methods for Time-Varying Convex Optimization.” in Proc. Asilomar Conf. on Signals Systems Computers, Pacific Grove, CA, November 8-11 2015.

<http://seas.upenn.edu/~akoppel/>