# Algorithm Design with Steering Vectors


Max Nilsson, PhD Student - Lund University
Department of Automatic Control
Supervisors: Pontus Giselsson (LU), Sebastian Banert (UBremen)

LUND UNIVERSITY


## Preliminaries

We aim to solve, over a Hilbert space $\mathcal{H}$, the monotone inclusion problem

> **Main Problem:** Find $\bar{x} \in \mathcal{H}$ such that $0 \in \sum_{i=1}^{m} A_i \bar{x}$.

where $A_i : \mathcal{H} \rightrightarrows \mathcal{H}$ is a maximally monotone operator. Our tools are the resolvents $J_{\gamma_i A_i} := (I + \gamma_i A_i)^{-1}$ for positive stepsizes $\gamma_i$ (resolvent splitting). This problem generalizes convex problems

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \sum_{i=1}^{m} f_i(x) \iff \underset{\bar{x} \in \mathbb{R}^n}{\text{Find}}\ 0 \in \sum_{i=1}^{m} \partial f_i(\bar{x}).$$

We suppose that there exists some solution of primal and dual variables $\bar{\chi} := (\bar{x}, \bar{y}_1, \ldots, \bar{y}_{m-1}) \in \mathcal{H}^m$ such that $\bar{y}_i \in A_i \bar{x}$ and $\bar{y}_m := -\sum_{i=1}^{m-1} \bar{y}_i \in A_m \bar{x}$.
The normal workflow of algorithm research:

$$\text{Algorithm Design} \longrightarrow \text{Lyapunov Analysis.}$$

But **how** did you come up with the algorithm and **why** does the Lyapunov analysis really work?

> **Our Workflow:** Algorithm Design $\longleftarrow$ Lyapunov Analysis.

We start with arbitrary points $\chi := (x, y_1, \ldots, y_{m-1}) \in \mathcal{H}^m$ and $\chi^+ := (x^+, y_1^+, \ldots, y_{m-1}^+) \in \mathcal{H}^m$ and a metric $\|\cdot\|_M$, where $M \in \mathbb{R}^{m \times m}$ is positive definite. Then, the inequality

$$\|\chi - \bar{\chi}\|_M \leq \|\chi^+ - \bar{\chi}\|_M \tag{1}$$

will drive the discussion, itself defining $\chi^+$ from $\chi$.

## Lyapunov Analysis

Let $\{z_i\}_{i=1}^m$ be a collection of points in $\mathcal{H}$ and $p_i := J_{\gamma_i A_i} z_i$.
Using the monotonicity of $A_i$ for $i \in [1, m-1]$ we get that

$$\left\langle \frac{z_i - p_i}{\gamma_i} - \bar{y}_i, p_i - \bar{x} \right\rangle \geq 0$$

and the monotinicity of $A_m$ we get that

$$\left\langle \frac{z_m - p_m}{\gamma_m} + \sum_{i=1}^{m-1} \bar{y}_i, p_m - \bar{x} \right\rangle \geq 0.$$

We get that (1) holds if the following inequality holds.

> **A Fundamental Quadratic Inequality:**
>
> $$\|\chi - \bar{\chi}\|_M - \|\chi^+ - \bar{\chi}\|_M + \lambda \left( \left\langle \frac{z_i - p_i}{\gamma_i} - \bar{y}_i, p_i - \bar{x} \right\rangle + \left\langle \frac{z_m - p_m}{\gamma_m} + \sum_{i=1}^{m-1} \bar{y}_i, p_m - \bar{x} \right\rangle \right) \leq 0$$
>
> where $\lambda \geq 0$.

Our first step will be to remove the *dependence* of the solution $\bar{\chi}$ from this inequality.

For simplicity, assume that $M = I$. After some work, we get that this fundamental inequality holds if and only if

$$\frac{1}{2}\|x^+ - x\|_2^2 + \langle x^+ - x, x \rangle + \sum_{i=1}^{m-1} \left( \frac{1}{2}\|y_i^+ - y_i\|_2^2 + \langle y_i^+ - y_i, y_i - \bar{y}_i \rangle \right)$$
$$+ \lambda \left\langle \frac{z_m - p_m}{\gamma_m} + \sum_{i=1}^{m-1} \bar{y}_i, p_m \right\rangle + \sum_{i=1}^{m-1} \lambda \left\langle \frac{z_i - p_i}{\gamma_i} - \bar{y}_i, p_i \right\rangle$$
$$- \left\langle x^+ - x + \lambda \sum_{i=1}^{m} \frac{z_i - p_i}{\gamma_i}, \bar{x} \right\rangle \leq 0$$

which implies that $x^+ := x - \lambda \sum_{i=1}^{m} (z_i - p_i)/\gamma_i$ *must hold*.
After similarly expanding the fundamental quadratic inequality, and using the definition of $x^+$, we get that $y_i^+ := y_i - \lambda(p_i - p_m)$ *must hold*.

## Algorithm Design

After performing the change of variables

$$\begin{cases} \tilde{z}_i &= z_i - x - \gamma_i y_i, \quad i = 1, \ldots, m-1 \\ \tilde{z}_m &= z_m - x + \gamma_m \sum_{i=1}^{m-1} y_i \\ \tilde{p}_i &= p_i - x \end{cases}$$

and using the definitions of $x^+$ and $y_i^+$ we get that the fundamental quadratic inequality holds if and only if

> **The Final Quadratic Inequality:**
>
> $$\mathbf{Q}(\tilde{\mathbf{z}}, \tilde{\mathbf{p}}, \lambda) := \frac{\lambda^2}{2} \left\| \begin{bmatrix} \sum_{i=1}^{m} (\tilde{z}_i - \tilde{p}_i)/\gamma_i \\ \tilde{p}_1 - \tilde{p}_m \\ \vdots \\ \tilde{p}_{m-1} - \tilde{p}_m \end{bmatrix} \right\|_2^2 + \lambda \sum_{i=1}^{m} \left\langle \frac{\tilde{z}_i - \tilde{p}_i}{\gamma_i}, \tilde{p}_i \right\rangle \leq 0.$$

This inequality we can factorize with the $LDL^\top$ factorization, which transforms the problem of *finding convergent algorithms* in $\mathcal{H}^m$ into an *eigenvalue problem* over $\mathbb{R}^{2m}$. Indeed, we can show that for $\lambda$ small enough, the signature of this quadratic form equals $(m, m, 0)$ which leads us to

> $$\sum_{i=1}^{m} d_i \Bigg\| \underbrace{\tilde{z}_i - \sum_{j=1}^{i-1} l_{i,j} \tilde{z}_j + l'_{i,j} \tilde{p}_j}_{=:v_i} \Bigg\|_2^2 \leq C$$

where the constants $d_i, l_{i,j}, l'_{i,j}$ and $C$ are precomputed from the $LDL^\top$ factorization, such that $d_i, C > 0$. The *steering vectors* $v_i$ can then be *designed* by the user to achieve faster convergence.

## Future Work

- Explore which algorithms we capture with this design method.
- Incorporate such that $x^+$ can depend on a longer history of iterations.
- Generalize this methodology so that it handles cocoercive operators.


WASP
WALLENBERG AI, AUTONOMOUS SYSTEMS AND SOFTWARE PROGRAM