PIPG for Rendezvous

Govind Chari

Univeristy of Washington

May 2023

PIPG Overview

PIPG

- Is a first order primal-dual conic optimizer
- Is easily verifiable
- Is factorization free
- Is an intuitive algorithm
- Is very fast (3-7x faster than ECOS)
- Can be coded in an afternoon
- Supports infeasibility detection

Comparison with ECOS

	ECOS	PIPG
Verification Ease	✗	✓
Factorization Free	✗	✓
Understandable	✗	✓
Implementation Ease	✗	✓
Infeasibility Detection	✓	✓
Tuning Free	✓	✗

PIPG Algorithm

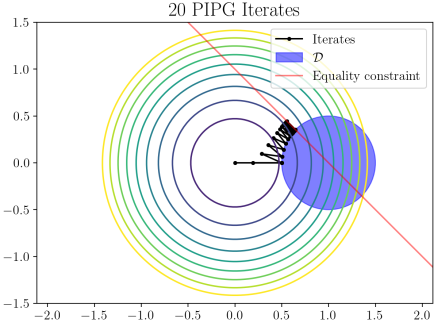
Optimization Problem

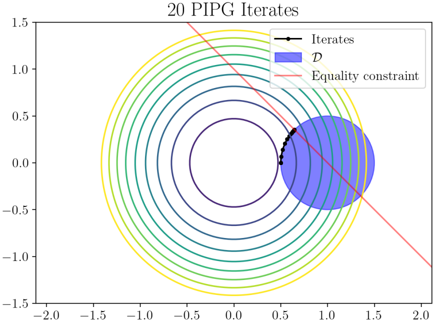
$$
\begin{aligned}
\min_{z} \quad & \frac{1}{2} z^\top P z + q^\top z \\
\text{s.t.} \quad & Hz - h = 0 \\
& z \in \mathbb{D}
\end{aligned}
$$

Algorithm

$$z^{k+1} \leftarrow \Pi_{\mathbb{D}} \left[z^k - \alpha (P z^k + q + H^\top w^k) \right]$$

$$w^{k+1} \leftarrow w^k + \beta (H (2 z^{k+1} - z^k) - h)$$

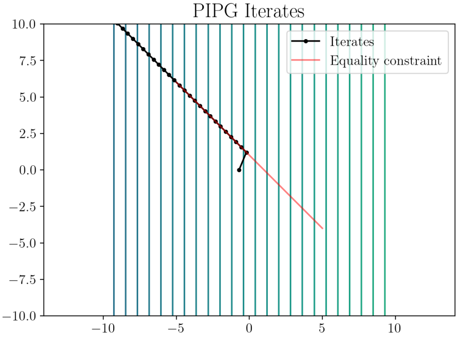


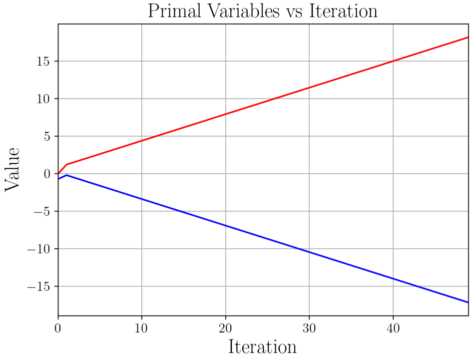


Infeasibility Detection - Dual Infeasibility

Dual Infeasibility Criteria:

$$\lim_{k \to \infty} \|z^{k+1} - z^k\| \neq 0$$

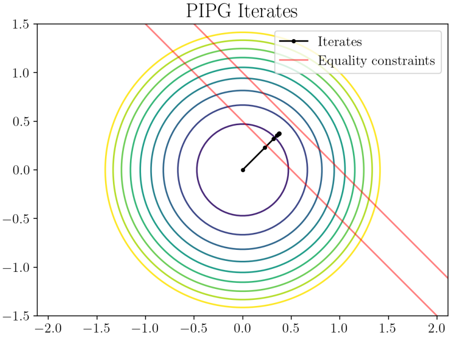


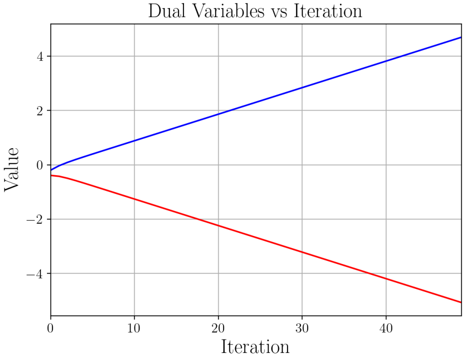


Infeasibility Detection - Primal Infeasibility

Primal Infeasibility Criteria:

$$\lim_{k \to \infty} \|w^{k+1} - w^k\| \neq 0$$

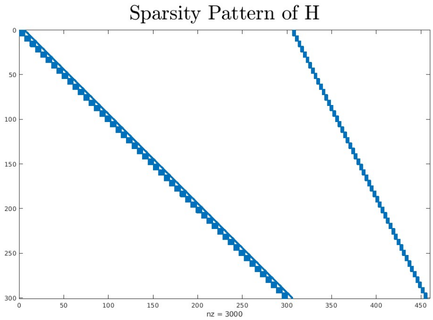


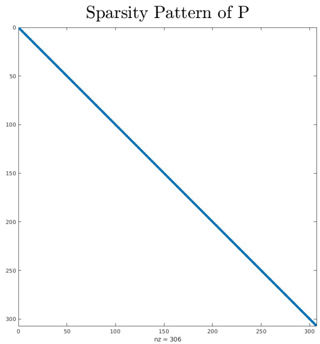


Structured Sparsity

$$\begin{aligned} \underset{x_t, u_t}{\text{minimize}} \quad & x_N^\top Q_N x_N + \sum_{t=1}^{N-1} x_t^\top Q_t x_t + u_t^\top R_t u_t \\ \text{subject to} \quad & x_{t+1} = A_t x_t + B_t u_t, \\ & x_t \in \mathbb{X}_t, \quad u_t \in \mathbb{U}_t \end{aligned}$$

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & \frac{1}{2} z^\top P z \\ \text{subject to} \quad & Hz - h = 0, \ z \in \mathbb{D} \end{aligned}$$

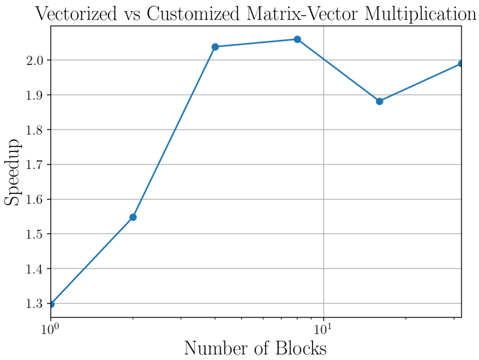




Customization: Structure Exploitation

$$\underbrace{\begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_K \end{bmatrix}}_{\text{H}} \underbrace{\begin{bmatrix} x_1 \\ \vdots \\ x_K \end{bmatrix}}_{\text{z}} = \underbrace{\begin{bmatrix} A_1 x_1 \\ \vdots \\ A_K x_K \end{bmatrix}}_{\text{y}}$$

We can compute $y = Hz$ subvector by subvector ($y_i = A_i x_i$) without constructing H and without using sparse-matrix vector multiplication.



Customization can double structured matrix vector multiplication speed.

Customization: PIPG

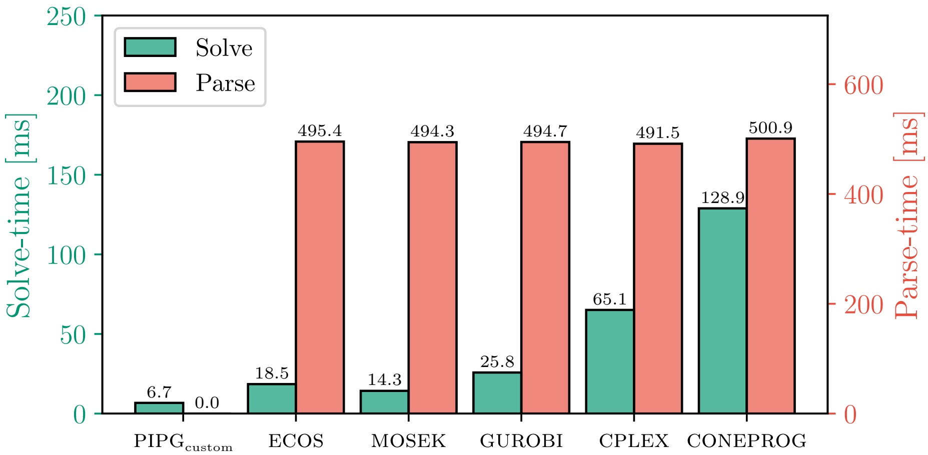
$$
\begin{aligned}
\underset{x_t, u_t}{\text{minimize}} \quad & x_N^\top Q_N x_N + \sum_{t=1}^{N-1} x_t^\top Q_t x_t + u_t^\top R_t u_t \\
\text{subject to} \quad & x_{t+1} = A_t x_t + B_t u_t, \\
& x_t \in \mathbb{X}_t, \quad u_t \in \mathbb{U}_t
\end{aligned}
$$

$$
\begin{aligned}
\underset{z}{\text{minimize}} \quad & \frac{1}{2} z^\top P z \\
\text{subject to} \quad & Hz - h = 0, \ z \in \mathbb{D}
\end{aligned}
$$

$$
\begin{aligned}
w_t &\leftarrow v_t + \beta(x_{t+1} - A_t x_t - B_t u_t) \\
u_t &\leftarrow \Pi_{\mathbb{U}_t}[u_t - \alpha(R_t u_t - B_t^\top w_{t-1})] \\
x_t &\leftarrow \Pi_{\mathbb{X}_t}[x_t - \alpha(Q_t x_t + w_{t-1} - A_t^\top w_t)] \\
v_t &\leftarrow v_t + \beta(x_{t+1} - A_t x_t - B_t u_t)
\end{aligned}
$$

$$
\begin{aligned}
w &\leftarrow v + \beta(Hz - h) \\
z &\leftarrow \Pi_{\mathbb{D}}[z - \alpha(Pz + H^\top w)] \\
v &\leftarrow v + \beta(Hz - h)
\end{aligned}
$$

Parsing

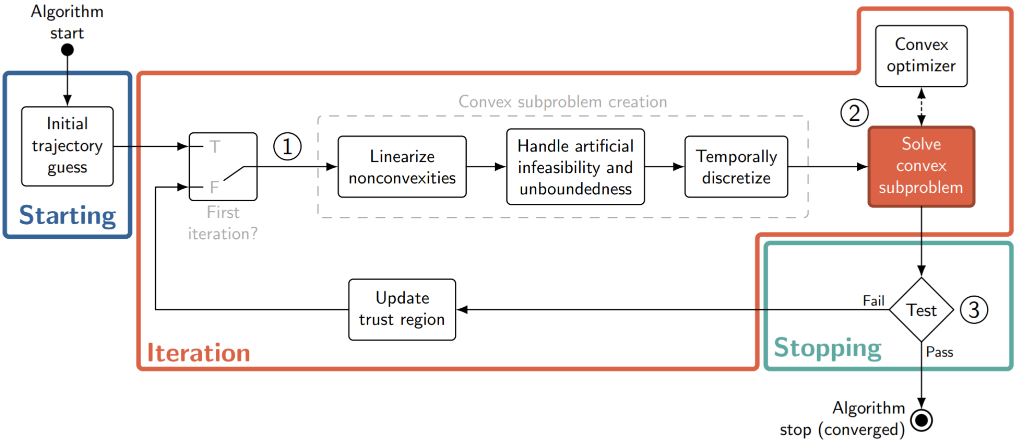


From Elango et al. *SciTech 2021*

Continuous-time Nonconvex Problem

$$
\begin{aligned}
\underset{t_f,\, x(t), u(t)}{\text{minimize}} \quad & \int_0^{t_f} \|u(t)\|_2^2 \, dt \\
\text{subject to} \quad & \forall t \in [0, t_f) \\
\boxed{\text{CW Dynamics}} \quad & \dot{x}(t) = f(x(t), u(t)) \\
\boxed{\text{Max delta-v}} \quad & \|u(t)\|_2 \leq u_{\text{max}} \\
\boxed{\text{Keepout Zone}} \quad & \|r(t) - r_c\|_2 \geq \rho \\
\boxed{\text{Max Speed}} \quad & \|v(t)\|_2 \leq v_{\max} \\
\boxed{\text{Initial Conditions}} \quad & r(0) = r_i \\
& v(0) = v_i \\
\boxed{\text{Terminal Conditions}} \quad & r(t_f) = 0_{3\times 1} \\
& v(t_f) = 0_{3\times 1}
\end{aligned}
$$

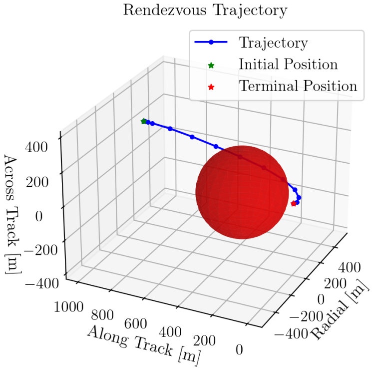
Successive Convex Programming (SCP)



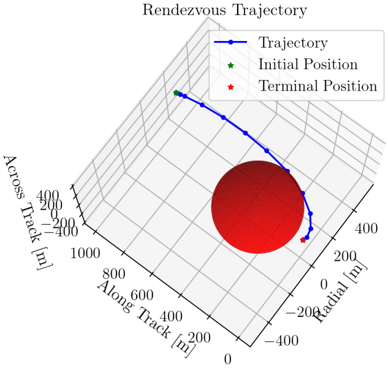
Discrete-time Convex Subproblem

$$
\begin{aligned}
\underset{\sigma, x, u}{\text{minimize}} \quad & \sum_{k=1}^{K-1} \|u_k\|_2^2 + w_{tr}\left(\sum_{k=1}^{K} \|x_k - \hat{x}_k\|_2^2 + \sum_{k=1}^{K-1} [\|u_k - \hat{u}_k\|_2^2 + (\sigma_k - \hat{\sigma}_k)^2]\right) + w_{vc}\|\nu^c\|_1 + w_{vb}\sum_{k=1}^{K} \nu_k^b \\
\text{subject to} \quad & \forall k \in [1, K] \\
\boxed{\text{Discrete Dynamics}} \quad & x_{k+1} = A_k x_k + B_k u_k + S_k \sigma_k + c_k + \nu_k^c \\
\boxed{\text{Dilation Constraints}} \quad & \sigma_{\text{min}} \le \sigma_k \le \sigma_{\text{max}} \\
\boxed{\text{Max delta-v}} \quad & \|u_k\|_2 \le u_{\text{max}} \\
\boxed{\text{Keepout Zone}} \quad & \|\hat{r}_k - r_c\|_2 + \left(\frac{\hat{r}_k - r_c}{\|\hat{r}_k - r_c\|_2}\right)^\top (r_k - \hat{r}_k) + \nu_k^b \ge \rho \\
& \nu_k^b \ge 0 \\
\boxed{\text{Max Speed}} \quad & \|v_k\|_2 \le v_{\max} \\
\boxed{\text{Initial Conditions}} \quad & r(0) = r_i \\
& v(0) = v_i \\
\boxed{\text{Terminal Conditions}} \quad & r(t_f) = 0_{3\times 1} \\
& v(t_f) = 0_{3\times 1}
\end{aligned}
$$

Trajectory

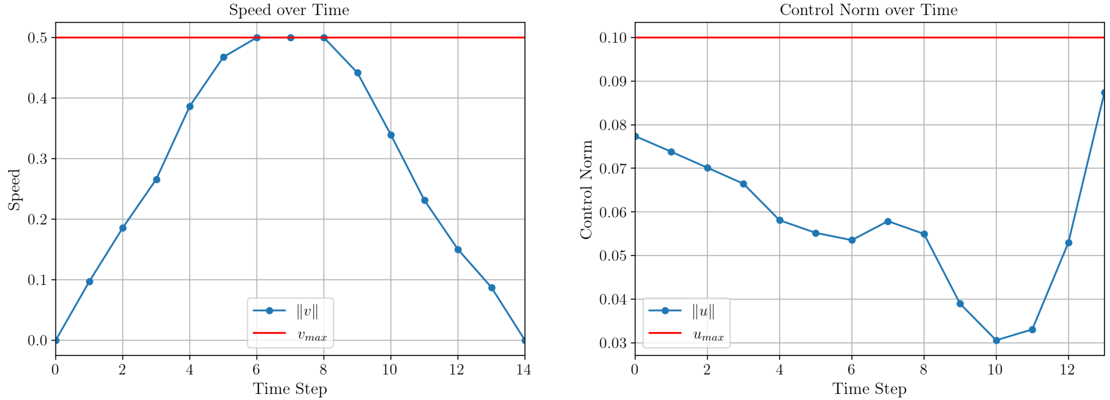


Rendezvous Trajectory
Trajectory
Initial Position
Terminal Position
Across Track [m]
400
200
0
−200
−400
1000
800
600
400
200
0
Along Track [m]
400
200
0
−200
−400
Radial [m]



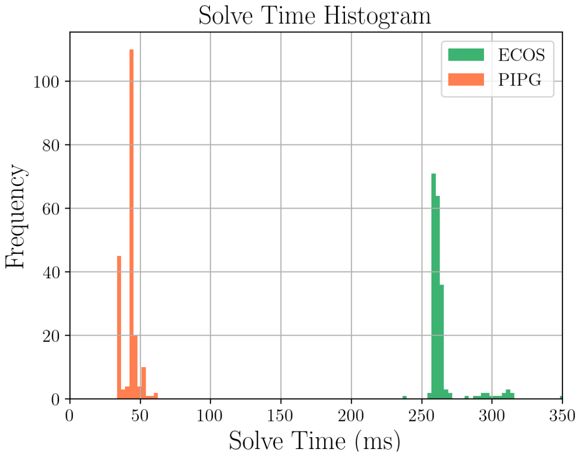
Rendezvous Trajectory
Trajectory
Initial Position
Terminal Position
Across Track [m]
400
200
0
−200
−400
1000
800
600
400
200
0
Along Track [m]
400
200
0
−200
−400
Radial [m]

Trajectory



Speed over Time
Speed
Time Step
0.0
0.1
0.2
0.3
0.4
0.5
0
2
4
6
8
10
12
14
$\|v\|$
v_{max}
Control Norm over Time
Control Norm
Time Step
0.03
0.04
0.05
0.06
0.07
0.08
0.09
0.10
0
2
4
6
8
10
12
$\|u\|$
u_{max}

Speed Results



200 runs, 3 millisecond bins

	ECOS	PIPG
Mean	261ms	**42ms**
Median	267ms	**44ms**
STD	23ms	**5ms**

PIPG is **6.25x** faster even without customization

Summary

PIPG

- Is easy to understand
- Is easy to verify
- Is easy to implement
- Is faster than ECOS