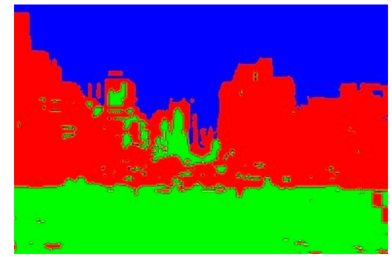
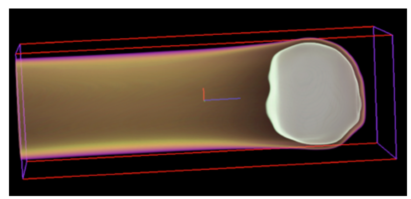
Motivations: Large Scale Problems

Several image processing applications may need solution of optimization problems

Pixel labeling:



Radiation therapy:



Super resolution:

These are large problems (10^6 pixels/voxels, …) with structure.

Hardware requires simple code but it is often parallel (GPU, SIMD)

Motivations: Optimization-based Control

Massively produced control algorithms are very basics, and microcontroller have very limited capabilities

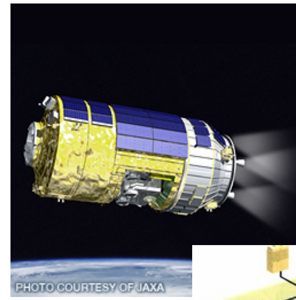
Realistic specs:
166 MHz, FP-capable
96 kB RAM,
2.5MB Data-ROM

PI(D) code versus MPC code (30% of it)

```
err = r - y ;
eint = eint + err ;
eder = (err – epre) / Ts ;
epre = err ;
u = Kp*err + Ki*eint + Kd*eder;
```

Code verification and validation is long and expensive.
Simple algorithms are always favored to complex ones.

Motivations: MPC in Mechatronics



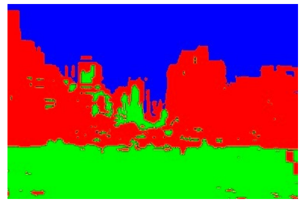
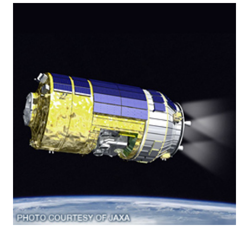
Fast systems (10Hz - 1KHz, …)
Low cost embedded CPU, small RAM, ROM
Unsupervised

⇔

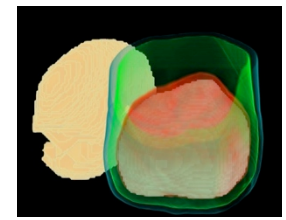
Fast execution.
Simple algorithm.
Verifiable code.

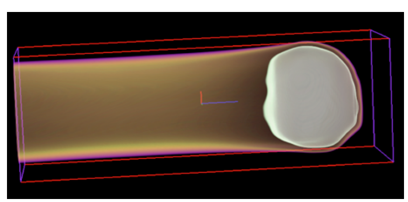
Need optimization algorithms that use few resources, run fast…
… and are simple to code and verify …

Algorithm Objectives

- Solve large problems
- Massive parallelization (image processing)
- Simple code (control system)
- “Fast” convergence



Outline

- **Multiplicative fixpoints for NNQP**
- Application to general QP
- Model predictive control
- Case studies

Optimality Conditions for NNQP

Consider the Non-Negative Quadratic Program (NNQP) $(H > 0)$

$$\begin{array}{ll} \min\limits_{z} & J(z) = \frac{1}{2} z' H z + F' z + M \\ \text{s.t.} & z \geq 0, \end{array}$$

with Lagrangian: $L(z, \lambda) = \frac{1}{2} z' H z + F' z - \lambda' z$

Under strict complementarity (assuming z>0, possibly infinitesimally) the optimality conditions are

$$z \circ \nabla_z L(z, \lambda) = 0 \qquad \boxed{z \circ (Hz + F - \lambda) = 0}$$

$$\lambda \circ \nabla_\lambda L(z, \lambda) = 0 \qquad \lambda \circ z = 0$$

$$[z]_i \geq 0, \ i \in \mathbb{Z}_{[1, n_z]}, \ [\lambda]_i \geq 0, \ i \in \mathbb{Z}_{[1, n_z]}$$

from variational methods (but also KKT under strict complementarity)

Cost Split and Fixpoint

Split the cost function in two parts in the optimality condition

$$z \circ \nabla_z L(z, \lambda) \quad = \quad z \circ ((H^+ z + F^+) - (H^- z + F^- + \lambda)) = 0.$$

Then, the optimal solutions are fixpoints of

$$z = \frac{H^- z + F^-}{H^+ z + F^+} \circ z$$

For active constraints: $[\lambda]_i > 0 \quad \Longrightarrow \quad [z]_i = 0$

For inactive constraints: $[\lambda]_i = 0 \quad \Longrightarrow \quad [(H^+ z + F^+)]_i = [(H^- z + F^-)]_i$

We design an algorithm to achieve such fixpoint without the need to estimate λ.

“Matrix Splitting Methods”, P. Tseng, SIAM 1991, …, Springer 2009
"Multiplicative updates for nonnegative quadratic programming”, Sha, et. al., , Neural Computation, 2007

Multiplicative Update Iteration

For any matrix ϕ

$$z \circ (Hz + F - \lambda) = z \circ ((H^+ z + F^+ + \phi z) - (H^- z + F^- + \lambda + \phi z)).$$

Starting from $z > 0$ we apply the multiplicative update

$$\boxed{z_{(h+1)} = \frac{(H^- + \phi)z + F^-}{(H^+ + \phi)z + F^+} \circ z_{(h)}}$$

where ϕ is a (usually small) constant matrix chosen to guarantee convergence.

Note: does not estimate λ (no need to…)

Proof of Convergence

If $\phi + \mathrm{diag}(F^+)\mathrm{diag}(z)^{-1} + H^- \geq 0$ for all $z > 0$,

$$z_{(h+1)} = \frac{(H^- + \phi)z_{(h)} + F^-}{(H^+ + \phi)z_{(h)} + F^+} \circ z_{(h)} \quad \Rightarrow \quad \lim_{z \to \infty} z_{(h)} = z^*$$

Proof strategy

Expand cost function : $J(\xi) = J(z) + (\xi - z)'\nabla J(z) + \frac{1}{2}(\xi - z)'H(\xi - z)$

Define the function : $\mathcal{G}(\xi, z) = J(z) + (\xi - z)'\nabla J(z) + \frac{1}{2}(\xi - z)'\mathcal{K}(z)(\xi - z)$

$$\mathcal{K}(z) = \mathrm{diag}\left(\frac{[(H^+ + \phi)z + F^+]_i}{[z]_i}\right)$$

1. $\forall \xi \geq 0, z > 0,\ \mathcal{G}(\xi, z) \geq J(\xi)$
2. $z_{(h+1)} = \arg\min_{\xi} \mathcal{G}(\xi, z_{(h)})$
3. $J(z_{(h+1)}) \leq \mathcal{G}(z_{(h+1)}, z_{(h)}) < \mathcal{G}(z_{(h)}, z_{(h)}) = J(z_{(h)})$
4. $\lim_{h \to \infty} z_{(h)} = z^*,\ \ z^* \geq 0$

Proof: Upper Bounding Function

Step 1.

For some non-negative symmetric matrix ϕ that depends only on H, $\mathcal{G}(\xi, z)$ for any $z > 0$ upper bounds $J(\xi)$, for any $\xi \geq 0$

Proof

$$\bar{H}^- \triangleq H^- + \phi$$
$$\bar{H}^+ \triangleq H^+ + \phi$$

$$\mathcal{K}(z) - H = \underbrace{(\mathrm{diag}(\bar{H}^+ z)\mathrm{diag}(z)^{-1} - \bar{H}^+)}_{\mathcal{K}_{psd}(z)} + \underbrace{(\mathrm{diag}(F^+)\mathrm{diag}(z)^{-1} + \bar{H}^-)}_{\mathcal{K}_{nn}(z)}$$

$$\mathcal{K}_{psd}(z) = \mathrm{diag}(\bar{H}^+ z)\mathrm{diag}(z)^{-1} - \bar{H}^+ \qquad \mathcal{K}_{nn}(z) = \mathrm{diag}(F^+)\mathrm{diag}(z)^{-1} + \bar{H}^-$$

For any symmetric nonnegative ϕ, $\mathcal{K}_{psd}(z) \geq 0$

So we can choose ϕ such that $\mathcal{K}_{nn}(z) \geq 0$

$$\boxed{\mathcal{K}(z) - H \geq 0}$$

E.g.: $\phi = \mathrm{diag}(H^- \mathbf{1})$

Proof: Minimization of Bounding Function

Step 2.

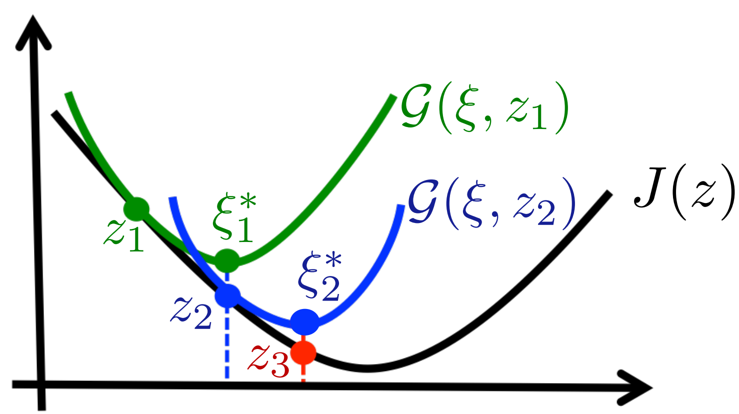
The value $\xi = \frac{(H^- + \phi)z + F^-}{(H^+ + \phi)z + F^+} \circ z$ minimizes $\mathcal{G}(\xi, z)$ for any $z > 0$

Proof

Optimality condition: $\nabla_\xi \mathcal{G}(\xi, z) = \nabla J(z) + \mathcal{K}(z)(\xi - z) = 0$

$$
\begin{aligned}
\xi &= z - \mathcal{K}(z)^{-1} \nabla J(z) \\
&= z - \mathcal{K}(z)^{-1}((H^+ + \phi)z + F^+) + \mathcal{K}(z)^{-1}((H^- + \phi)z + F^-) \\
&= \mathrm{diag}(z)\mathrm{diag}((H^+ + \phi)z + F^+)^{-1}((H^- + \phi)z + F^-).
\end{aligned}
$$

Also, any $z > 0$ results in $\xi \geq 0$



Proof: Monotonic Cost Decrease

Step 3.

Given $z_{(h)} \neq z^*$, if $\exists i: \; [z]_{i(h)}[(H^+ + \phi)z_{(h)} + F^+]_i \neq 0$:

$$J(z_{(h+1)}) \leq \mathcal{G}(z_{(h+1)}, z_{(h)}) < \mathcal{G}(z_{(h)}, z_{(h)}) = J(z_{(h)})$$

Proof

We have convexity and strict convexity for *i-th* variable

$$\frac{\partial^2}{\partial [z_{(h+1)}]_i^2} \mathcal{G}(z_{(h+1)}, z_{(h)}) = [\mathcal{K}(z_{(h)})]_{ii} = \frac{[(H^+ + \phi)z_{(h)} + F^+]_i}{[z_{(h)}]_i} > 0 \;.$$

We have $[z_{(h+1)}]_i = [\arg\min_\xi \mathcal{G}(\xi, z_{(h)})]_i \neq [z_{(h)}]_i$

$$[z_{(h+1)}]_i - [z_{(h)}]_i = -[\mathcal{K}(z_{(h)})^{-1} \nabla J(z_{(h)})]_i \neq 0$$

Proof: Convergence to Optimum

Step 4.

Given $z_{(0)} > 0$ the sequence generated by the multiplicative update is such that

$$\lim_{h\to\infty} J(z_{(h)}) = J^* \qquad \lim_{h\to\infty} z_{(h)} = z^*$$

Proof

$$[z_{(h+1)}]_i - [z_{(h)}]_i = -[\mathcal{K}(z_{(h)})^{-1}\nabla J(z_{(h)})]_i \neq 0$$

$z_{(h)} = z_{(h+1)} \Leftrightarrow z \circ \nabla J(z) = 0$ optimality condition for NNQP

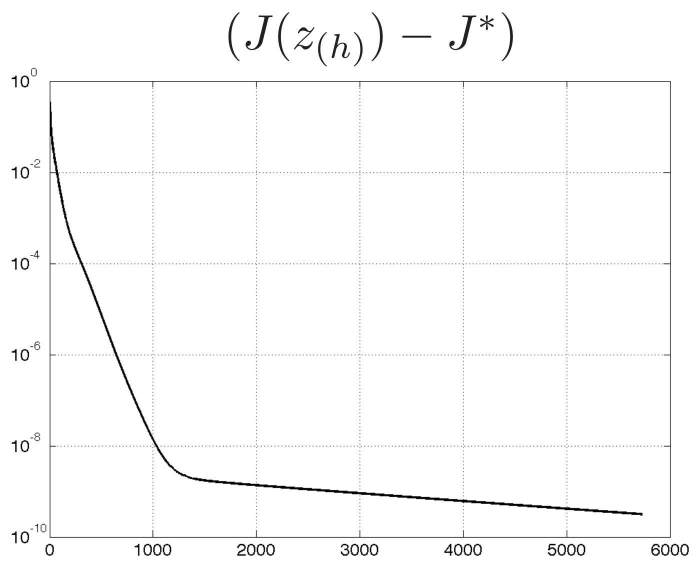
$z > 0 \implies \nabla J(z) = 0$ and by convexity $\nabla J(z) = 0 \Leftrightarrow z = z^*$

$\lim_{h\to\infty} \{J(z_{(h)})\}_h = \bar{J}$ but the only stationary point is z^*

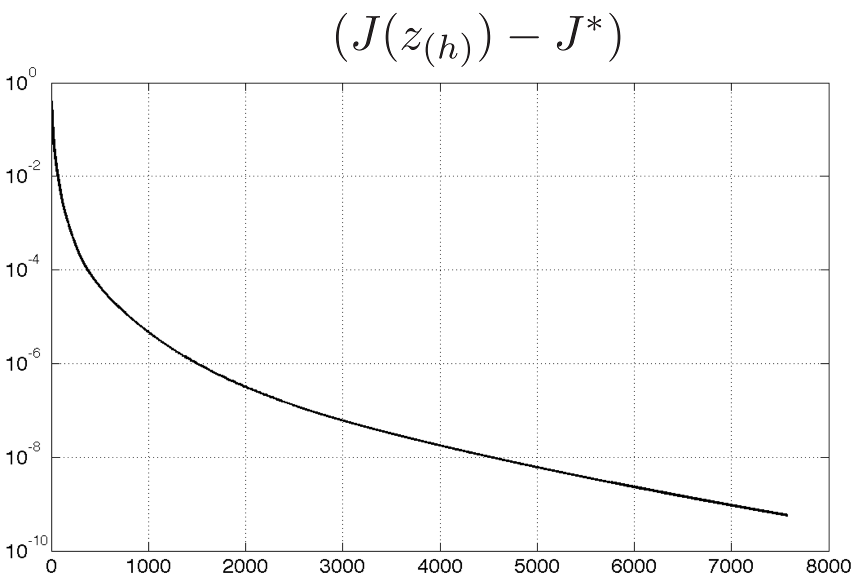
$$\lim_{h\to\infty} \{J(z_{(h)})\}_h = \bar{J} = J^*$$

Examples: Cost Sequences

Random NNQP



60 variables



1000 variables

Multiplicative Update: Properties

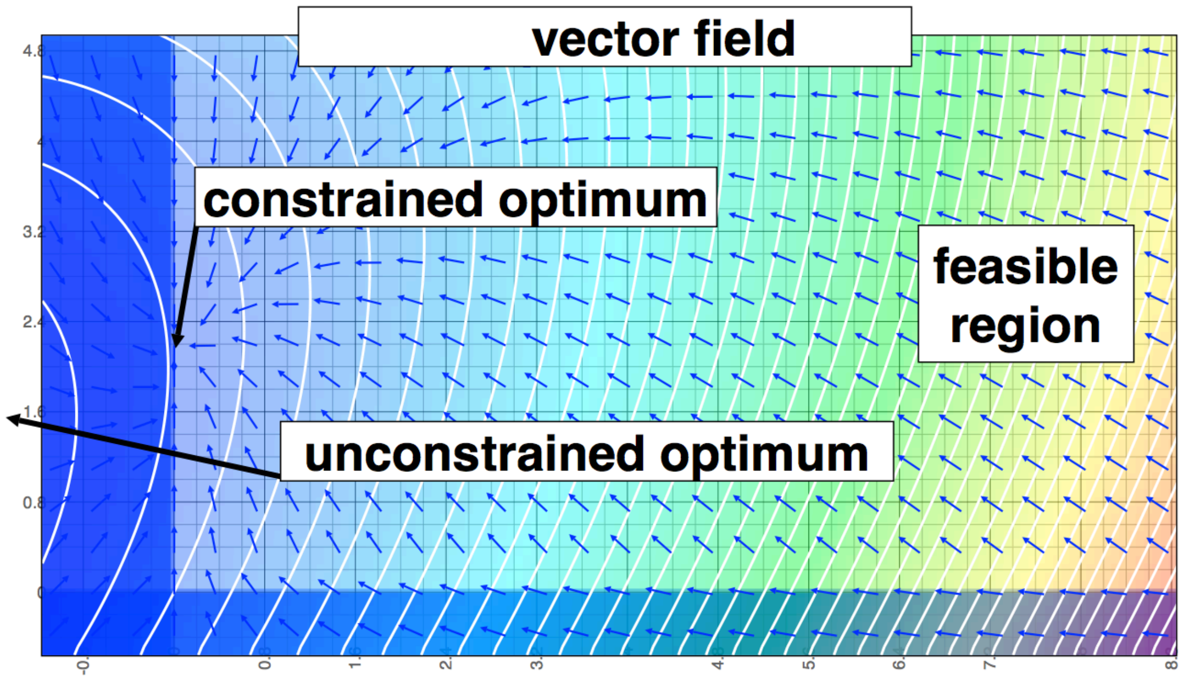
Iteration update (simple ☺): $$[z_{(h+1)}]_i = \frac{[(H^- + \phi)z_{(h)} + F^-]_i}{[(H^+ + \phi)z_{(h)} + F^+]_i}[z_{(h)}]_i$$

1. **Projection:** not needed since if $z_{(h)} > 0$ then $z_{(h+1)} > 0$ ☺
2. **Monotonicity:** cost sequence is monotonic $J(z_{(h+1)}) < J(z_{(h)}), \forall h \in \mathbb{R}_{0+}$ ☺
3. **Convergence rate:** linear convergence can be proved (for $H > 0$)
4. **Complexity order:** $O(n_z^2 \cdot p)$ versus $O(n_z^3 \cdot \log(p))$ for general I.P. methods
   p = # precision digits
5. **Operation types:** sums, multiplications ☺, divisions ☹
   (numerator-denominator: 0-complementary)
6. **Parallelizability:** easy due to the update ☺
   (can exploit matrix bandwidth)

The algorithm is called **Parallel Quadratic Programming (PQP)** because of 6 (and by 1, PQP could be also “Projection-free QP”).

Multiplicative Update: Vector Field

The multiplicative update defines a (static) vector field on the feasible domain

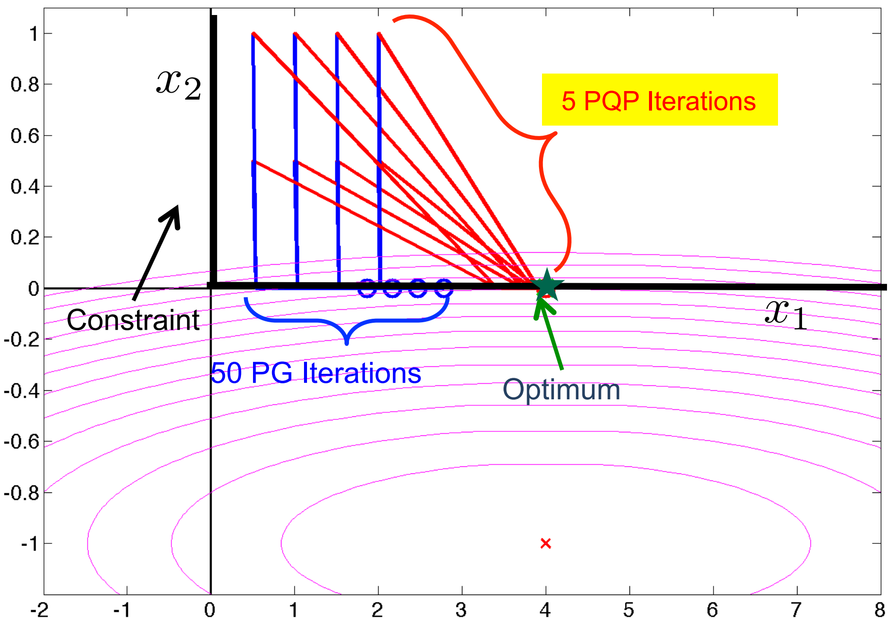


the denominator acts as a "barrier" that repels from the border of the feasible set except in a manifold leading to the optimum

Properties: Projection-free

The multiplicative update does not need an (explicit) projection.

Although inexpensive in NNQP projection can deteriorate performance…



The PQP iteration does not need to apply a *try-and-correct* procedure (= *step-and-project*)

Properties: Asynchronous Update

Only a subset of variables can be updated

$$[\xi]_i = \left[\frac{(H^- + \phi)z + F^-}{(H^+ + \phi)z + F^+}\right]_i \circ [z]_i$$

which is valuable for single-thread multirate architecture (e.g., control system)

Remember that the convergence proof works by:

$$\text{if } \exists i: \ [z]_{i(h)}((H^+ + \phi)z_{(h)} + F^+)_i \neq 0:$$

$$[z_{(h+1)}]_i = [\arg\min_\xi \mathcal{G}(\xi, z_{(h)})]_i \neq [z_{(h)}]_i$$

$$[z_{(h+1)}]_i - [z_{(h)}]_i = -[\mathcal{K}(z_{(h)})^{-1}\nabla J(z_{(h)})]_i \neq 0$$

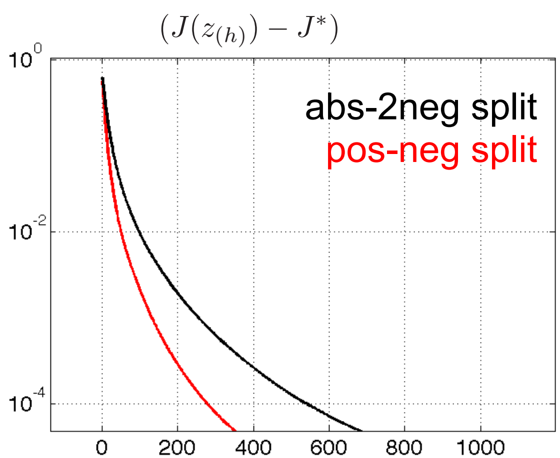
Alternative Cost Function Splits

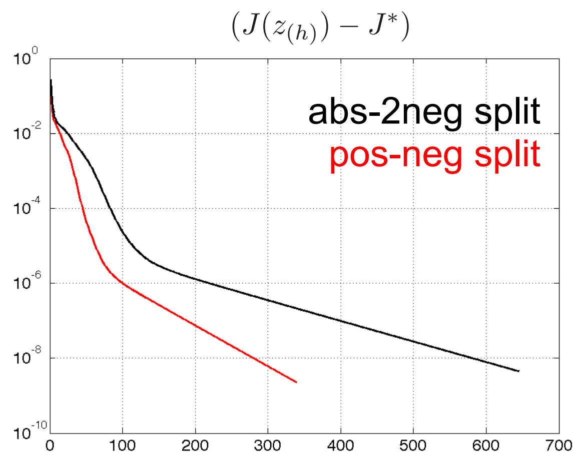
The choice of cost function split is not unique

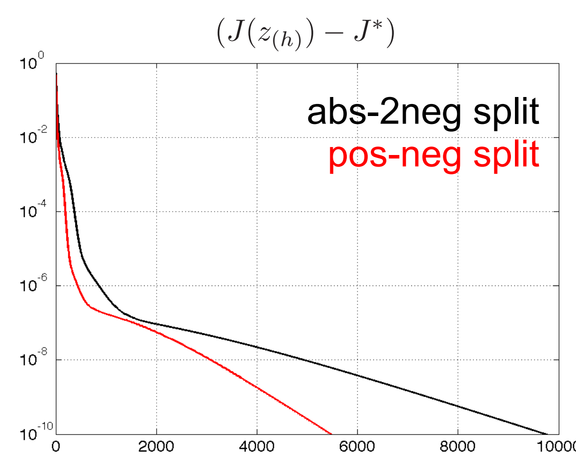
Absolute value splitting: $z \circ \nabla_z L(z, \lambda) \quad = \quad z \circ ((|H|z + F^+) - (2H^- z + F^- + \lambda))$

$$\boxed{\text{Multiplicative update: } z_{(h+1)} = \frac{(2H^- + \phi)z + F^-}{(|H| + \phi)z + F^+} \circ z_{(h)}}$$

Often is "slower" than pos-neg split, but sometimes it is more ``robust"







Convergence Rate

Assume there exists $0 < R_1 \leq R_2$: $R_1\|z\|^2 \leq z'Hz \leq R_2\|z\|^2$, $\forall z \geq 0,\ z \neq 0$

For $\{z_{(h)}\}_h,\ z_{(h)} > 0,\ \forall h \in \mathbb{Z}_{0+}$, there exists $\rho > 0$ such that $\frac{\|z_{(h+1)}-z^*\|}{\|z_{(h)}-z^*\|} \leq \rho$

Proof sketch

From convergence proof: $[z_{(h+1)}]_i - [z_{(h)}]_i = -[\mathcal{K}(z_{(h)})^{-1}\nabla J(z_{(h)})]_i$

Then we have: $\frac{\|z_{(h+1)}-z^*\|}{\|z_{(h)}-z^*\|} \leq \rho = \max(|1-m_{(h)}|, |1-M_{(h)}|)$

$$m_{(h)} = \lambda_{\min}(\mathcal{K}(z_{(h)})^{-1/2}H\mathcal{K}(z_{(h)})^{-1/2}) \qquad M_{(h)} = \lambda_{\max}(\mathcal{K}(z)^{-1/2}H\mathcal{K}(z)^{-1/2})$$

and furthermore: $\rho \leq \max\left\{1 - \min_i\left\{\frac{R_1[z_{(h)}]_i}{[H^+z_{(h)}+F]_i}\right\}, \max_i\left\{\frac{[F]_i^+}{[H^+z_{(h)}+F]_i}\right\}\right\}$

$$\rho \leq \max\left\{1 - \min_{i\in\mathcal{I}(z^*)}\left\{\frac{\bar{R}_1[z^*_{(h)}]_i}{[H^+z^*_{(h)}+F]_i}\right\}, \max_{i\in\mathcal{A}(z^*)}\left\{\frac{[F]_i^+}{[H^+z^*_{(h)}+F]_i}\right\}\right\}$$

see also, Jian at al, Operators and matrices 2009

Outline

- Multiplicative fixpoints for NNQP
- **Application to general QP**
- Model predictive control
- Case studies

Application to General Convex QP

In several applications (control) we need to solve the general (convex) QP

$$\begin{aligned} \min_{U} \quad & J_p(U) = \frac{1}{2}U'Q_pU + F_p'U + \frac{1}{2}M_p \\ \text{s.t.} \quad & G_pU \leq K_p \end{aligned}$$

The dual is the NNQP:

$$\begin{aligned} \min_{Y} \quad & J_d(Y) = \frac{1}{2}Y'Q_dY + F_dY + \frac{1}{2}M_d \\ \text{s.t.} \quad & Y \geq 0 \end{aligned}$$

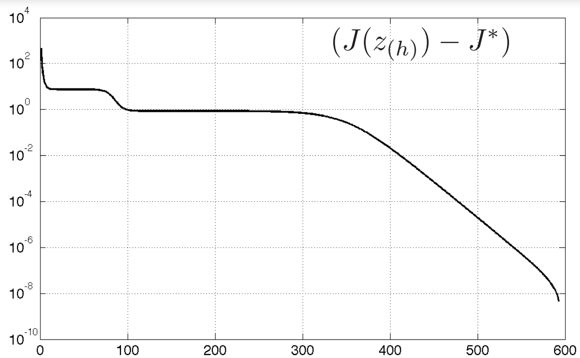
$$Q_d = G_pQ_p^{-1}G_p',\ F_d = (K_p + G_pQ_p^{-1}F_p)$$

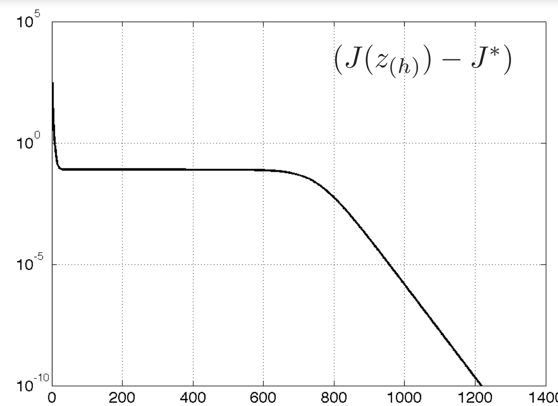
Approach: solve the dual and recover the primal

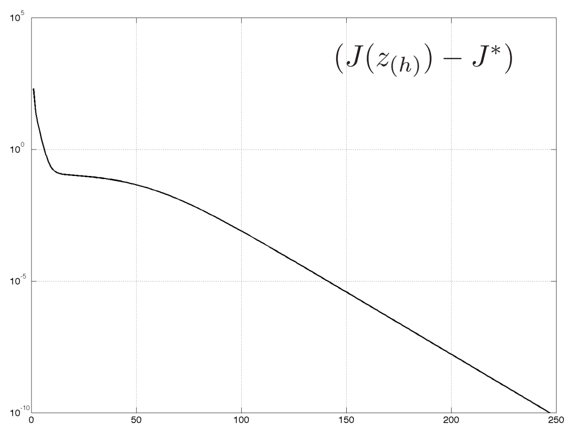
$$U^* = \psi_{d2p}(Y^*) = -Q_p^{-1}(F_p + G_pY^*)$$

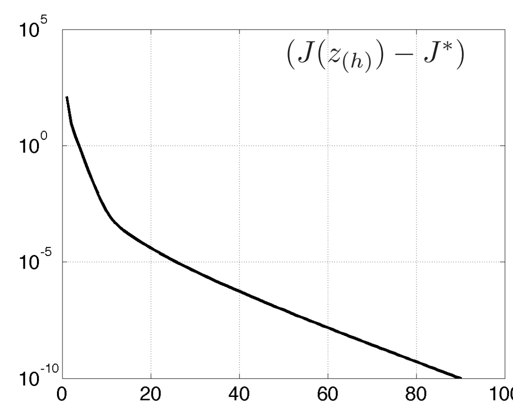
Convergence for General Convex QPs

The dual NNQP of a primal QP is (in our applications) weakly convex.
Convergence still holds but … there *may be* long “almost-flat” updates…







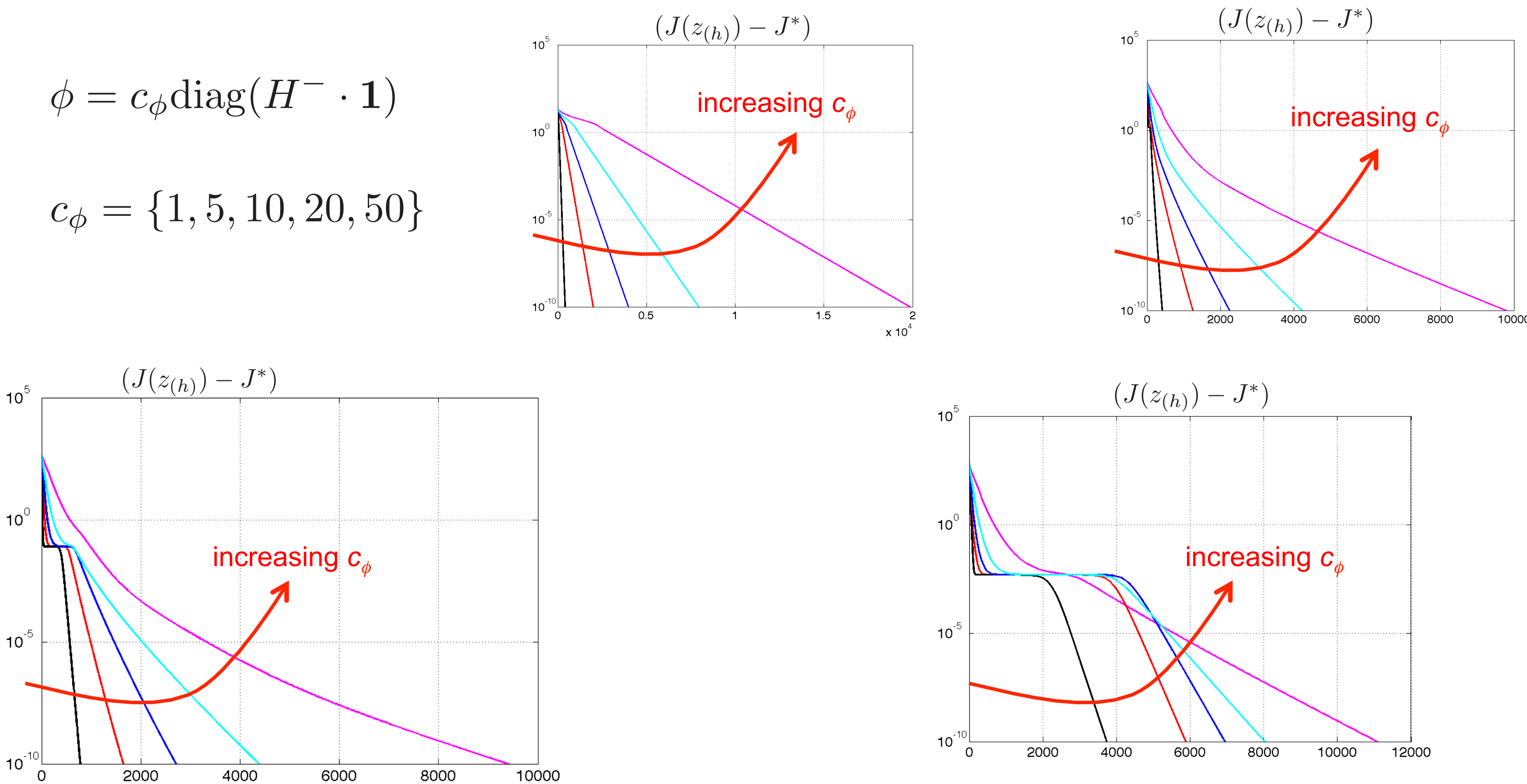


Dependence on Convexification Parameter

In general the "magnitude" of ϕ reduces the speed of convergence
However, it may reduce "almost-flat" updates

$$\phi = c_\phi \mathrm{diag}(H^- \cdot \mathbf{1})$$

$$c_\phi = \{1, 5, 10, 20, 50\}$$

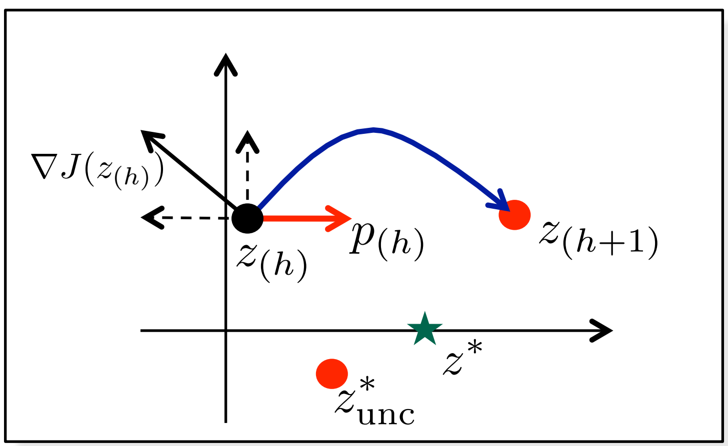


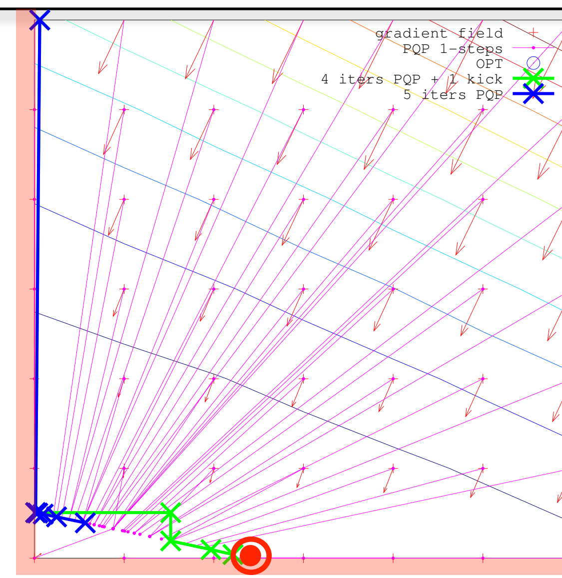
PQP with LGNC-Step ("Kick")

We accelerate convergence by "perturbing" the sequence of iterates.

We interleave the PQP iteration with optimal line search step along the gradient directions pointing towards the interior of the nonnegative cone

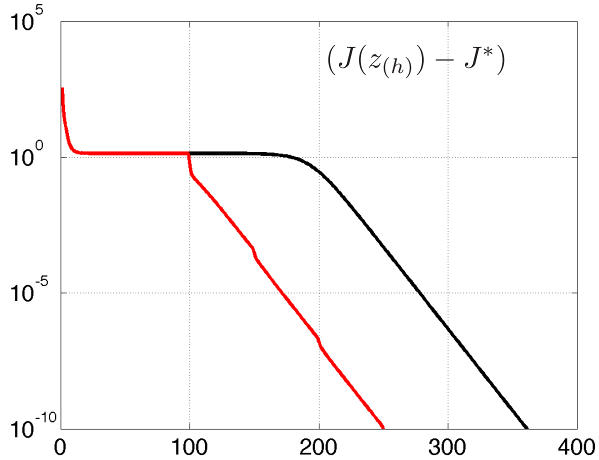
$$
\begin{aligned}
p_h &= (\nabla_z J(z_{(h)}))^- \\
\alpha(z_{(h)}) &= \begin{cases} -\frac{\nabla_z J(z_{(h)})' p_h}{p_h' H p_h} & \text{if } p_h' H p_h > 0 \\ 0 & \text{otherwise} \end{cases} \\
z_{(h+1)} &= z_{(h)} + \alpha(z_{(h)}) p_h,
\end{aligned}
$$

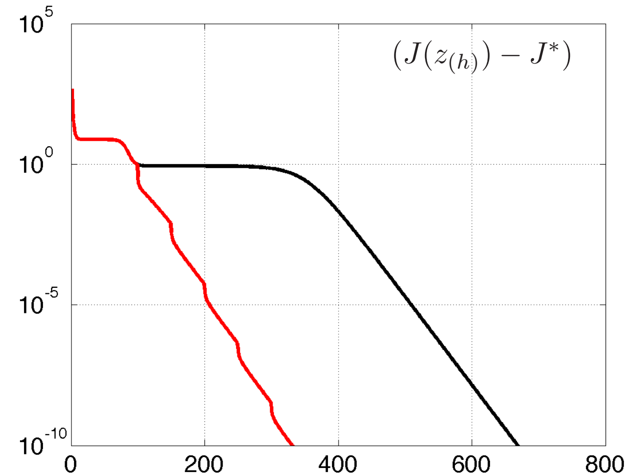


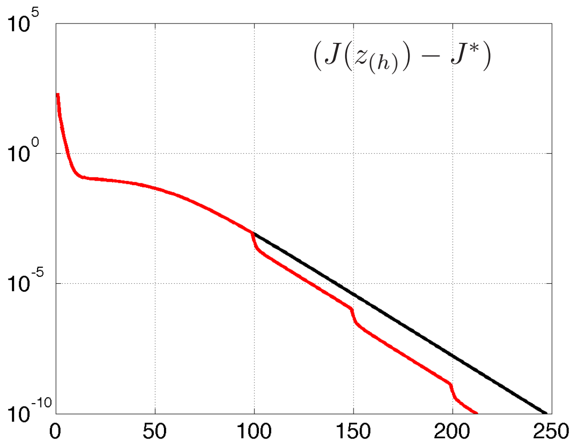


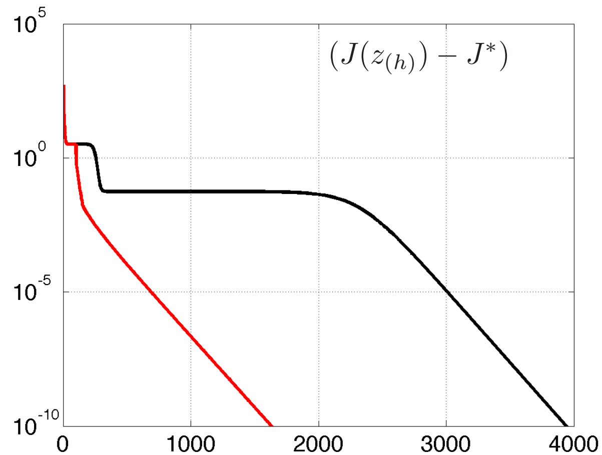
Periodic “Kick” Effects

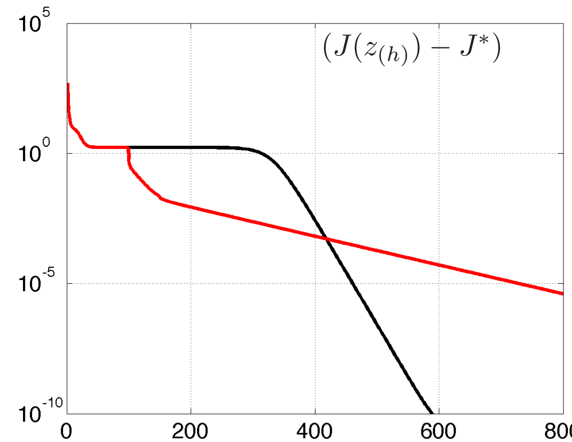
Every *n=50* iterations apply a LGNC update











Why is the "Kick" Helping?

PQP can be interpreted as an adaptive stepsize gradient method

$$[z_{(h+1)}]_i = [z_{(h)}]_i - \frac{[z_{(h)}]_i}{[H^+ z_{(h)} + F^+]_i} [Hz_{(h)} + F]_i,$$

$$z_{(h+1)} \quad = \quad z_{(h)} - \mathcal{K}(z_{(h)})^{-1} \nabla J(z_{(h)})$$

The scaling preserves feasibility and it is intrinsic in the iteration update.
Due to nonlinearity it may happen that the iteration "slows down"

Note: the scaling guarantees feasibility of the step at the price of losing the stepsize as a decision variable -> good for simplicity, but not good for avoiding to slow

Condition-triggered "Kick" strategy

We can design specific update selection strategies, e.g.:

- chose always the (local) best between PQP and LGNC
- LGNC when several non-optimal coordinates do not contribute to the update

$$|[\mathcal{K}(z)^{-1}\nabla J(z)]_i| < \epsilon \quad \text{and} \quad \begin{cases} [\nabla J(z)]_i > 0 \rightarrow \text{may be satisfying LC} \\ [\nabla J(z)]_i = 0 \rightarrow \text{at optimum} \\ \boxed{[\nabla J(z)]_i < 0 \rightarrow \text{not satisfying LC}} \end{cases}$$

$$\Downarrow$$

$$[z_{(h+1)}]_i - [z_{(h)}]_i < \frac{\epsilon^2}{|[Hz_{(h)} + F]_i|}$$

$$\Downarrow$$

GNC update

Outline

- Multiplicative fixpoints for NNQP
- Application to general QP
- **Model predictive control**
- Case studies

LQ-Model Predictive Control and QPs

Control the linear system, subject to state, output, input constraints:

$$\begin{aligned} x(k+1) &= Ax(k)+Bu(k) \\ y(k) &= Cx(k)+Du(k) \end{aligned} \qquad \begin{aligned} x_{\min} \leq &\; x(k) \leq x_{\max} \\ u_{\min} \leq &\; u(k) \leq u_{\max} \\ y_{\min} \leq &\; y(k) \leq y_{\max} \end{aligned}$$

Given $x(k)$, LQ-MPC solves at every iteration :

$$\begin{aligned} \min_{U_k} \quad & \|x_{N|k}\|^2_{P_{\mathrm{M}}} + \sum_{i=0}^{N-1} \|x_{i|k}\|^2_{Q_{\mathrm{M}}} + \|y_{i|k}\|^2_{S_{\mathrm{M}}} + \|u_{i|k}\|^2_{R_{\mathrm{M}}} \\ \text{s.t.} \quad & x_{i+1|k} = Ax_{i|k} + Bu_{i|k} \\ & y_{i|k} = Cx_{i|k} + Du_{i|k} \\ & \quad x_{\min} \leq x_{i|k} \leq x_{\max},\ i \in \mathbb{Z}_{[1,N]} \\ & \quad u_{\min} \leq u_{i|k} \leq u_{\max},\ i \in \mathbb{Z}_{[0,N-1]} \\ & \quad y_{\min} \leq y_{i|k} \leq y_{\max},\ i \in \mathbb{Z}_{[0,N-1]} \\ & x_{0|k} = x(k), \end{aligned}$$

by converting it into the QP:

$$\begin{aligned} \min_{U_k} \quad & J_p(U_k) = \frac{1}{2}U_k' Q_p U_k + F_p' U_k + \frac{1}{2}M_p \\ \text{s.t.} \quad & G_p U_k \leq K_p \end{aligned}$$

and then applies: $u(k) = u^*_{0|k}$

PQP Application to Model Predictive Control

When applied to MPC the QP is in state-parametric form

$$\begin{aligned} \min_{U} \quad & \frac{1}{2}U'Q_pU + x'C_p'U + \frac{1}{2}x'\Omega_p x \\ \text{s.t.} \quad & G_pU \leq S_px + W \end{aligned}$$

The dual problem can also be computed in parametric form

$$\begin{aligned} \min_{U} \quad & \frac{1}{2}U'Q_pU + x(k)'C_p'U + \frac{1}{2}x'\Omega_p x \\ \text{s.t.} \quad & G_pU \leq S_px(k) + W \end{aligned}$$

and so is the dual-primal optimum transformation

$$U(Y^*) = \Psi_{d2p}(x, Y^*) = \Gamma_d x(k) + \Xi_d Y^*$$

Termination conditions (primal feasibility + ε−suboptimality, absolute and relative)

$$-S_dx - W_d - Q_dY \leq \max\{\varepsilon_c^r(|S_px + W_p|), \varepsilon_c^a\mathbf{1}\}$$

$$Y'Q_dY+(x(k)'S_d'+W_d)Y \leq \max\left\{-\varepsilon_J^r\frac{1}{2}(Y'Q_dY + 2(x(k)'S_d + W_d)'Y + x(k)'\Omega_d x), \varepsilon_J^a\right\}.$$

PQPMPC: Complete Algorithm

$k = 0$
loop
  $h = 0,\ Y_{(h)} = \bar{Y} > 0.$
  repeat
    if LGNC condition (e.g., $\mathrm{mod}(h, \mathrm{LSrate})$) **then**

$$
\begin{aligned}
p_h &= (Q_d Y_{(h)} + x(k)' S_d + W_d)^- \\
\alpha(Y_{(h)}) &= \begin{cases} -\frac{(Q_d Y_{(h)} + x(k)' S_d + W_d)' p_h}{p_h' Q_d p_h} & \text{if} \quad p_h' Q_d p_h > 0 \\ 0 & \text{otherwise} \end{cases} \\
Y_{(h+1)} &= Y_{(h)} + \alpha(Y_{(h)}) p_h
\end{aligned}
$$

    else
      for $i = 1 : n_d$ **do**

$$
[Y_{(h+1)}]_i = \frac{[(Q_d^- + \phi)Y_{(h)} + (x(k)' S_d' + W_d)^-]_i}{[(Q_d^+ + \phi)Y_{(h)} + (x(k)' S_d' + W_d)^+]_i} [Y_{(h)}]_i,
$$

      end for
    end if
    $h = h + 1$
  until $-S_d x(k) - W_d - Q_d Y \le \max\{\ldots\}$
    and $Y' Q_d Y + (x(k)' S_d' + W_d) Y \le \max\{\ldots\}$
  set $u(k) = \begin{bmatrix} I_m & 0 & \ldots & 0 \end{bmatrix} (\Gamma_d x(k) + \Xi_d Y)$
  $k = k + 1$
end loop

initialize

LGNC-update

PQP-update

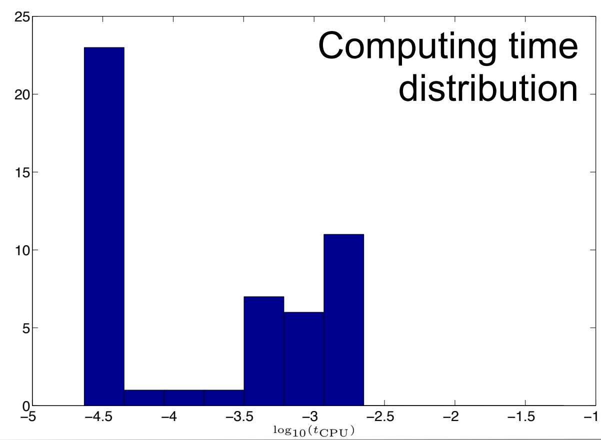
termination, input selection

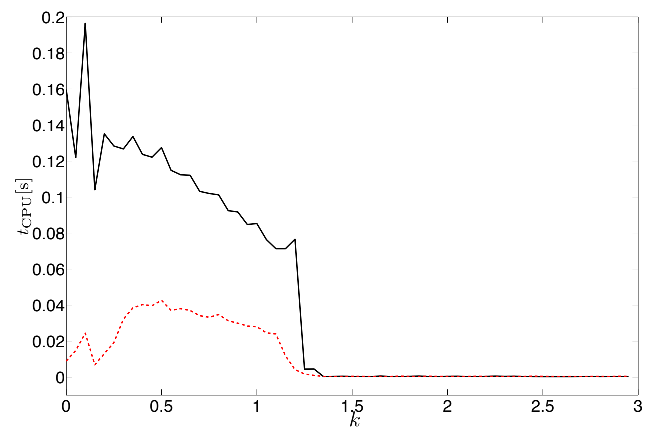
Benchmark: Aircraft Pitch and AoA Control

Track pitch and angle of attack of a jet aircraft.

4^{th} order system, 2 inputs
output and input constraints
5 steps horizon,
32 constraints, 10 variables

Solver	Avg[ms]	Min[ms]	Max[ms]
GPADM:	46.567	0.222	196.481
PQPM:	11.515	0.319	42.618
QUADPROG:	2.954	1.462	7.948
QPACT:	0.597	0.445	0.986
NAG:	0.917	0.615	1.387
PQPMEX:	0.937	0.069	3.553
PQPMPC:	0.444	0.032	1.985



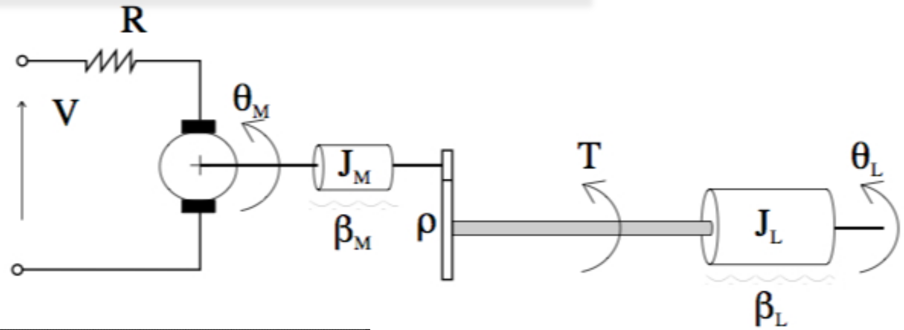


Computing time: **PQP** and **Dual Fast Gradient**

Benchmark: Servomotor Position Control

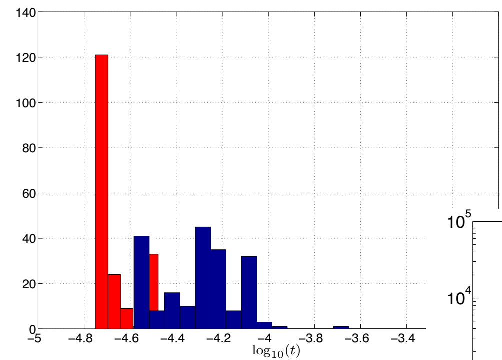
Track the reference position of the load connected to servo by flexible shaft with torsional and voltage constraints

4th order system,1 input
output and input constraints
10 steps horizon,
40 constraints, 10 variables



Mild reference

Solver	Avg[ms]	Min[ms]	Max[ms]
GPADM:	9.481	0.268	43.356
PQPM:	1.418	0.367	8.891
QUADPROG:	1.847	1.406	4.082
QPACT:	0.571	0.452	0.896
NAG:	0.865	0.599	1.637
PQPMEX:	0.178	0.077	0.987
PQPMPC:	0.084	0.040	0.491

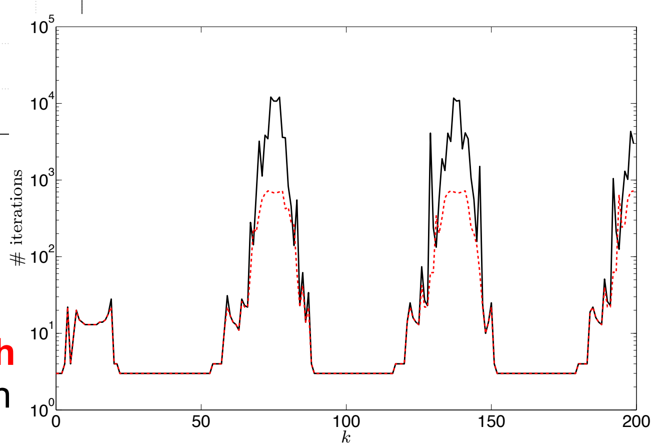


Computing time for mild and aggressive references

Aggressive reference

Solver	Avg[ms]	Min[ms]	Max[ms]
GPADM:	38.896	0.270	127.560
PQPM:	8.771	0.371	68.309
QUADPROG:	2.363	1.489	7.551
QPACT:	0.644	0.451	2.525
NAG:	0.852	0.371	1.669
PQPMEX:	0.727	0.078	6.469
PQPMPC:	0.367	0.040	3.037

Number of iterations **with** and **without** acceleration



Outline

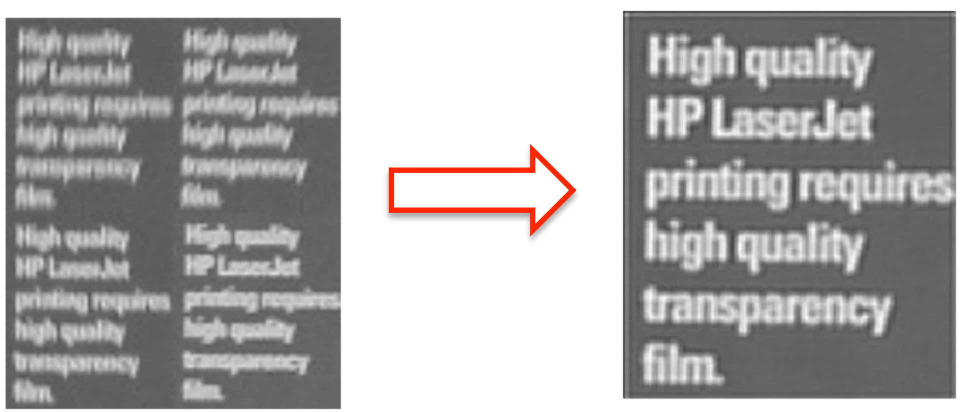
- Multiplicative fixpoints for NNQP
- Application to general QP
- Model predictive control
- **Case studies**

Image Processing Applications: Super-Resolution

Reconstruct a high-resolution image from many low resolution ones

$$\begin{array}{ll} \min & \frac{1}{2}\sum_{k=1}^{K} \|D_k S_k f - g_k\|^2 \\ \text{s.t.} & f \geq 0 \end{array}$$

Reconstruct a 285x245 image from 30 images 57x49



Brand, Chen, IEEE ICIP2011

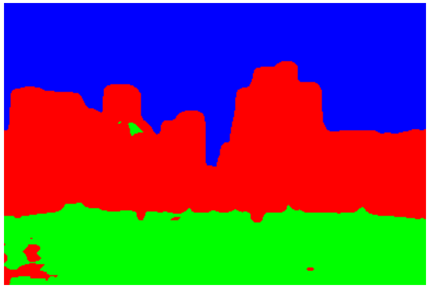
Image Processing Applications: Segmentation

Interactive segmentation: a user labels few pixels and an algorithm propagates the labels to the rest of the image.
MRF-based-> propagates probabilities

$$\begin{aligned} \min \quad & \frac{1}{2}\sum_{k=1}^{K}\sum_{r\in\Omega}\frac{\eta}{2}\sum_{s\in\mathcal{N}(r)} \omega_{rs}(x_k(r)-x_k(s))^2 + d_{rk}x_k(r) \\ \text{s.t.} \quad & x_k(r) \geq 0 \\ & \sum_{k=1}^{K} x_k(r) = 1 \end{aligned}$$

solve by the dual or multiplicative update on augmented Lagrangian

$O(10^6)$ variables, 5secs

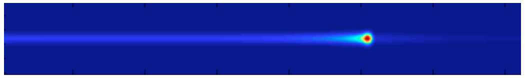


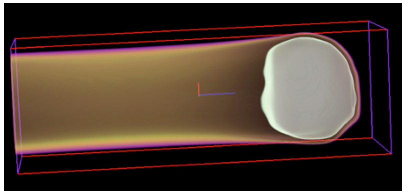
Brand, Chen, IEEE ICIP2011

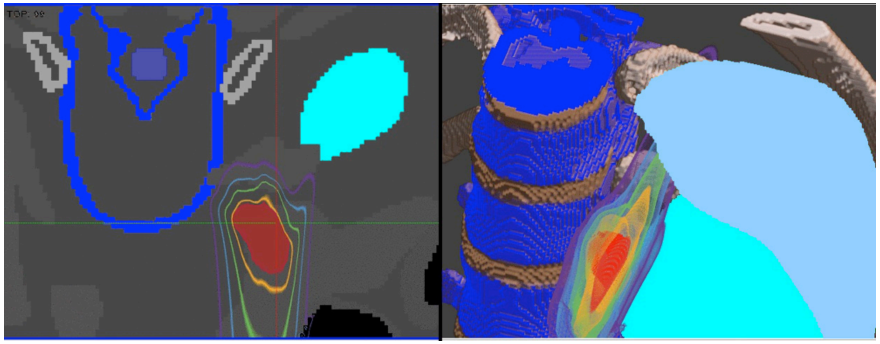
Biomedical Applications: Radiation Therapy

Plan the treatment by targeting beams in the appropriate areas to hit the entire tumor and avoid to hit the healthy tissue and vital organs.

$$\begin{array}{ll} \min\limits_{u} & u'Wu \\ \text{s.t.} & b \leq Au \leq c \end{array} \qquad \begin{array}{l} u \approx O(10^5), \\ b, c \approx O(10^6), \\ A \approx O(10^{11}) \end{array}$$







Superimpose $O(10^4\text{-}10^5)$ beams to irradiate tumors in an $O(10^6)$ voxel volume:

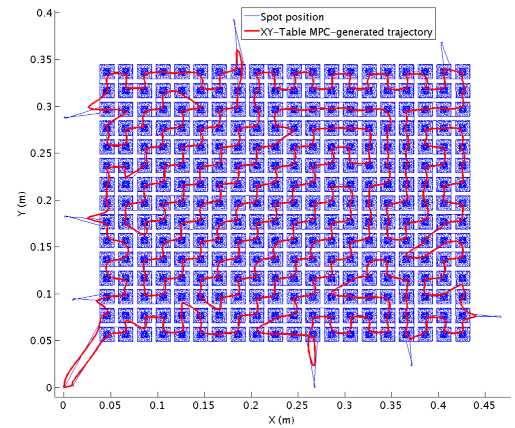
- Every tumor voxel has a minimum required dose (covering)
- Every voxel has a unique maximum acceptable dose (packing)
- Minimize excess radiation (L2-norm minimization)

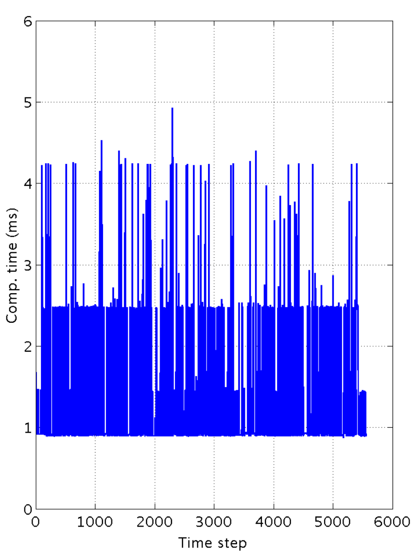
Conventional approaches: 20 minutes to 8 hours. PQP: < 3 seconds

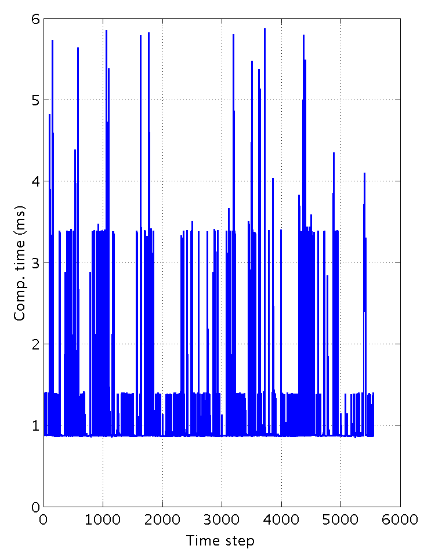
Control Applications: MPC for Laser Processing

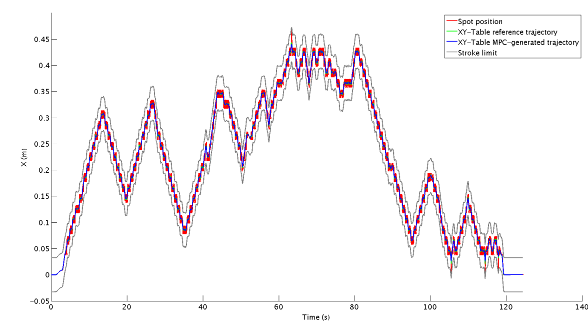
Coordinated control of a (constrained) redundantly actuated laser processing machine

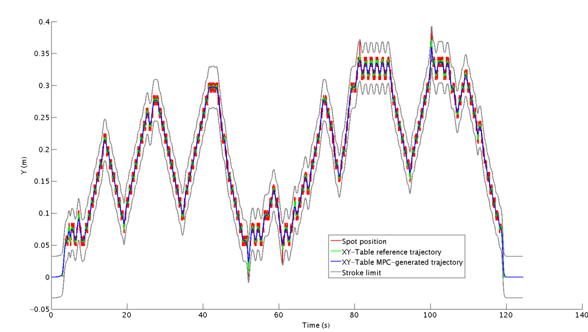
Update rate 25ms,
Prediction horizon 1s
40 variables,
588 constraints (2x)







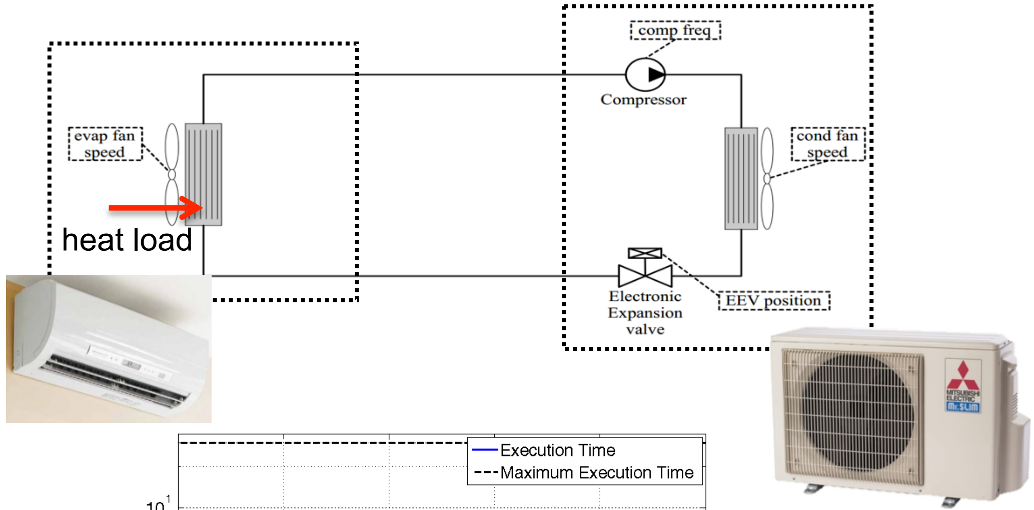




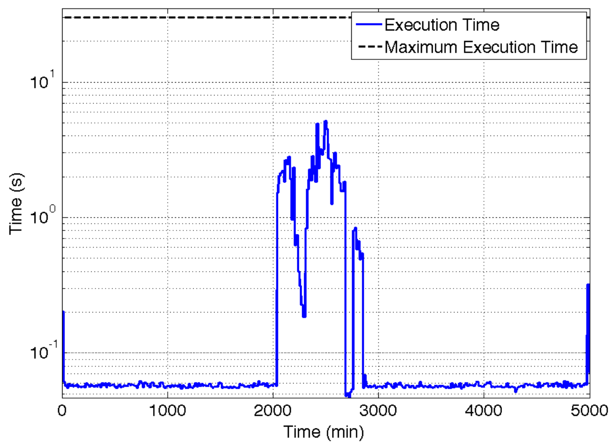
Haghighat, SDC, Konobrytsky, Bortoff, ASME DSCC 2014

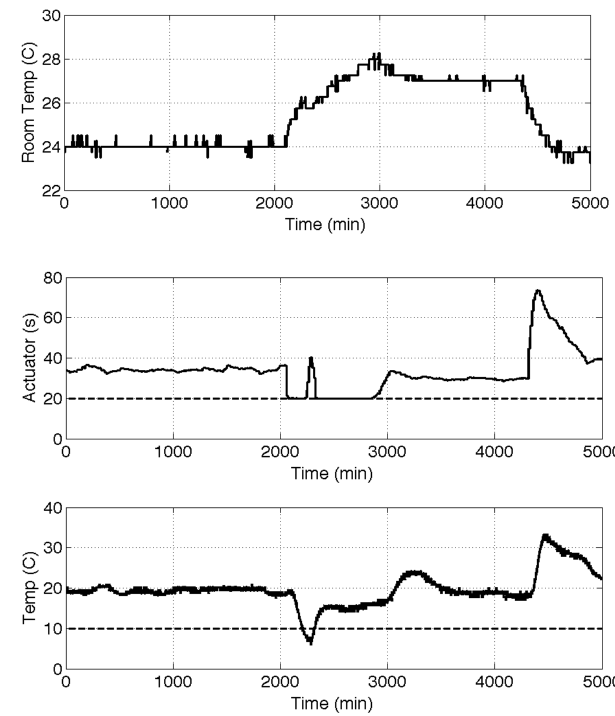
Real World Applications: MPC for HVAC control

Multivariable constrained control of compressor, valve, and fan in a room air conditioner



Update rate 30s,
103 variables, 513 constraints





Jain, Burns, SDC, Laughman, Bortoff, Refrigeration 2014

Conclusions and Future Work

Cost function splits results in multiplicative updates with guaranteed convergence.
We obtain algorithms that are simple to code and verify with fast convergence
Performance is (at least) comparable to other methods with similar code complexity.
Multiplicative update seems to provide fast convergence to near-optimum.

Current and future research:
- more detailed study on convergence rate
- bound on number of operations
- better characterization of properties of different splits
- parallelization & implementation in different architectures

References **(available at www.merl.com)**

[1] Brand, M., Chen, D., "Parallel Quadratic Programming for Image Processing", IEEE International Conference on Image Processing (ICIP), 2011
[2] Di Cairano, S., Brand, M., Bortoff, S.A., "Projection-free Parallel Quadratic Programming for Linear Model predictive Control", International Journal of Control, 2013.
[3] Di Cairano, S., Brand, M., "On a Multiplicative Update Dual Optimization Algorithm for Constrained Linear MPC", IEEE Conference on Decision and Control, 2013.
[4] Sha, F., Lin, Y., Saul, L.K., Lee, D.D., "Multiplicative updates for nonnegative quadratic programming." Neural Computation, 2007
[5] P. Tseng et. al., various works on “Matrix splitting methods”, SIAM 1991,…,Springer 2009

Workshop on Embedded Optimization,
September 9th, 2014, Lucca, Italy

MITSUBISHI ELECTRIC RESEARCH LABORATORIES
Cambridge, Massachusetts

Multiplicative fixpoints for convex quadratic programs

M. Brand**, S. Di Cairano*

*Mechatronics, MERL

**Algorithms, MERL

Collaborating MERL Interns ('10-'13):
M. Livne (U.Toronto), D. Chen (Tufts)
N. He (GaTech.), D. Meng (U.Wash)