

Model Predictive Control and Reinforcement Learning

– Lecture 07: Real-Time Algorithms for NMPC –

Moritz Diehl

University of Freiburg

Slides jointly developed with Armin Nurkanović, Leonard Fichtner, Florian Messerer, Katrin Baumgärtner,
Andrea Ghezzi, Jonathan Frey

School on Model Predictive Control and Reinforcement Learning
Freiburg, 7-16 October 2026

universität freiburg



NTNU

Norwegian University of
Science and Technology



Nonlinear MPC and Nonlinear Programming Problems (NLP)

Newton-Type Optimization and Sequential Convex Programming Methods

Real-Time Iterations

Nonlinear MPC and Nonlinear Programming Problems (NLP)

Newton-Type Optimization and Sequential Convex Programming Methods

Real-Time Iterations

Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Discrete time NMPC Problem (an NLP)

$$\begin{aligned} \min_{x,u} \quad & \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = f(x_k, u_k) \\ & 0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ & 0 \geq r(x_N) \end{aligned}$$

Variables $x = (x_0, \dots, x_N)$ and $u = (u_0, \dots, u_{N-1})$ can be summarized in vector $w = (x, u) \in \mathbb{R}^n$.

Nonlinear Program (NLP)

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & F(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & H(w) \geq 0 \end{aligned}$$

The Q-Loss of Model Predictive Control (MPC)

$$\begin{aligned} Q_{\text{MPC}}(s, a) &= \min_{x, u} \sum_{k=0}^{N-1} \ell(x_k, u_k) + E(x_N) \\ \text{s.t. } &x_0 = s \\ &u_0 = a \\ &x_{k+1} = f(x_k, u_k) \\ &0 \geq h(x_k, u_k), \quad k = 0, \dots, N-1 \\ &0 \geq r(x_N) \end{aligned}$$

The MPC feedback law is fully equivalent to $\pi_{\text{MPC}}(s) = \arg \min_a Q_{\text{MPC}}(s, a)$. Thus, MPC can be regarded a way to approximate the Q-function, which is based on a deterministic model and online optimization over a prediction horizon.

Nonlinear MPC and Nonlinear Programming Problems (NLP)

Newton-Type Optimization and Sequential Convex Programming Methods

Real-Time Iterations



Regard NLP with both, equalities and inequalities:

NLP with equality and inequality constraints

$$\min_w F(w) \quad \text{s.t.} \quad \begin{cases} G(w) = 0 \\ H(w) \geq 0 \end{cases}$$

Lagrangian function for NLP with equality and inequality constraints

$$\mathcal{L}(w, \lambda, \mu) = F(w) - \lambda^\top G(w) - \mu^\top H(w)$$

Theorem (Karush-Kuhn-Tucker (KKT) conditions)

Let F, G, H be $\mathcal{C}^2$. If w^* is a (local) minimizer and satisfies LICQ, then there are unique vectors λ^* and μ^* such that (w^*, λ^*, μ^*) satisfies:

$$\nabla_w \mathcal{L}(w^*, \mu^*, \lambda^*) = 0$$

$$G(w^*) = 0$$

$$H(w^*) \geq 0$$

$$\mu^* \geq 0$$

$$H(w^*)^\top \mu^* = 0$$

- ▶ These contain nonsmooth conditions (the last three) which are called *complementarity conditions*
- ▶ This system cannot be solved by Newton's Method. But still with a "generalized Newton's method" ...

Sequential Quadratic Programming (SQP)

By Linearizing all functions within the KKT Conditions, and setting $\lambda^+ = \lambda^k + \Delta\lambda$ and $\mu^+ = \mu^k + \Delta\mu$, we obtain the KKT conditions of a Quadratic Program (QP) (we omit these conditions).

QP with inequality constraints

$$\begin{aligned} \min_{\Delta w} \quad & \nabla F(w^k)^\top \Delta w + \frac{1}{2} \Delta w^\top A^k \Delta w \\ \text{s.t.} \quad & \begin{cases} G(w^k) + \nabla G(w^k)^\top \Delta w = 0 \\ H(w^k) + \nabla H(w^k)^\top \Delta w \geq 0 \end{cases} \end{aligned}$$

with

$$A^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k, \mu^k)$$

and its solution delivers

$$\Delta w^k, \quad \lambda_{\text{QP}}^+, \quad \mu_{\text{QP}}^+$$

Constrained Gauss-Newton Method

In special case of least squares objectives (tracking, parameter estimation)

Least squares objective function

$$F(w) = \frac{1}{2} \|R(w)\|_2^2$$

can approximate Hessian $\nabla_w^2 \mathcal{L}(w^k, \lambda^k, \mu^k)$ by much cheaper "Gauss-Newton Hessian approximation" (no multipliers needed):

$$A_{\text{Gauss-Newton}}^k := \nabla R(w) \nabla R(w)^\top.$$

QP= linear least squares:

Gauss-Newton QP

$$\begin{aligned} \min_{\Delta w} \quad & \frac{1}{2} \|R(w^k) + \nabla R(w^k)^\top \Delta w\|_2^2 \\ \text{s.t.} \quad & G(w^k) + \nabla G(w^k)^\top \Delta w = 0 \\ & H(w^k) + \nabla H(w^k)^\top \Delta w \geq 0 \end{aligned}$$

Convergence: linear (better if $\|R(w^*)\|$ small)

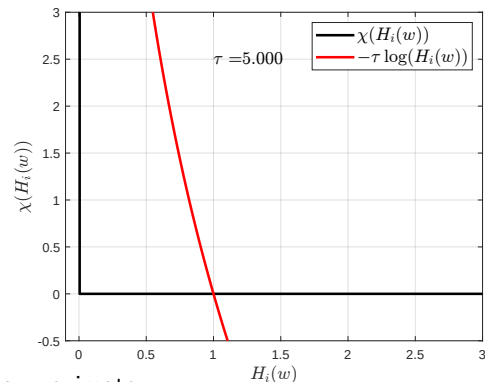
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \\ & G(w) = 0 \end{aligned}$$

Barrier problem

$$\begin{aligned} \min_w \quad & F(w) - \tau \sum_{i=1}^m \log(H_i(w)) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Idea: put inequality constraint into objective via "logarithmic barrier"



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

The barrier problem in interior point methods

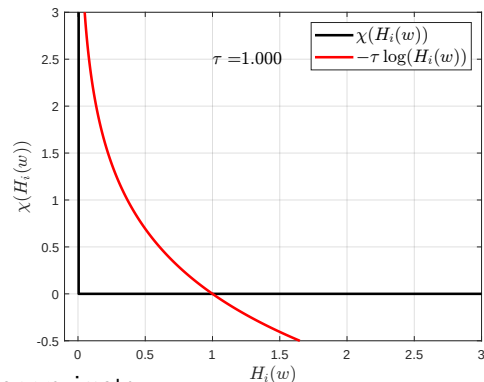
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \\ & G(w) = 0 \end{aligned}$$

Barrier problem

$$\begin{aligned} \min_w \quad & F(w) - \tau \sum_{i=1}^m \log(H_i(w)) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Idea: put inequality constraint into objective via "logarithmic barrier"



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

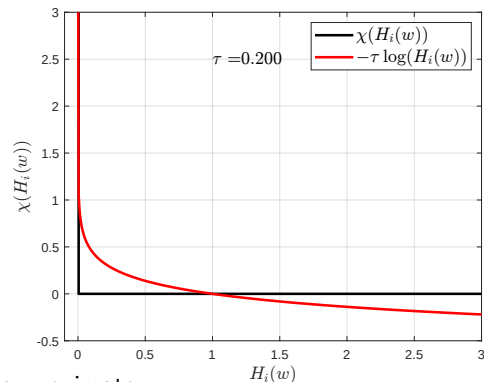
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \\ & G(w) = 0 \end{aligned}$$

Barrier problem

$$\begin{aligned} \min_w \quad & F(w) - \tau \sum_{i=1}^m \log(H_i(w)) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Idea: put inequality constraint into objective via "logarithmic barrier"



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

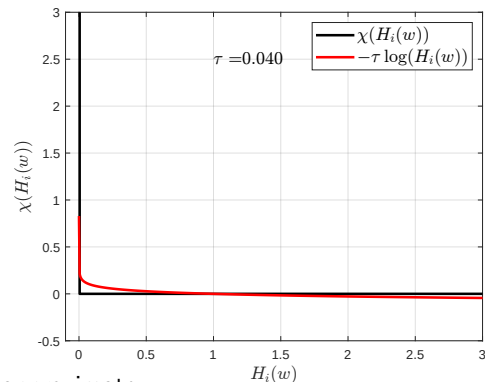
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \\ & G(w) = 0 \end{aligned}$$

Barrier problem

$$\begin{aligned} \min_w \quad & F(w) - \tau \sum_{i=1}^m \log(H_i(w)) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Idea: put inequality constraint into objective via "logarithmic barrier"



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

The barrier problem in interior point methods

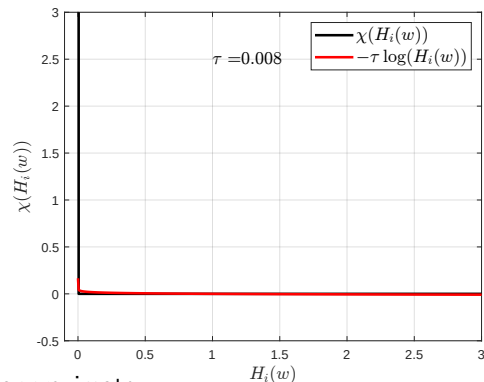
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \\ & G(w) = 0 \end{aligned}$$

Barrier problem

$$\begin{aligned} \min_w \quad & F(w) - \tau \sum_{i=1}^m \log(H_i(w)) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Idea: put inequality constraint into objective via "logarithmic barrier"



approximate:

$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

The barrier problem in interior point methods

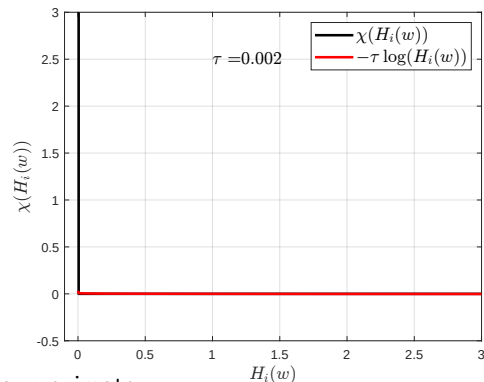
NLP with inequalities

$$\begin{aligned} \min_w \quad & F(w) \\ \text{s.t.} \quad & H(w) \geq 0 \\ & G(w) = 0 \end{aligned}$$

Barrier problem

$$\begin{aligned} \min_w \quad & F(w) - \tau \sum_{i=1}^m \log(H_i(w)) \\ \text{s.t.} \quad & G(w) = 0 \end{aligned}$$

Idea: put inequality constraint into objective via "logarithmic barrier"



approximate:

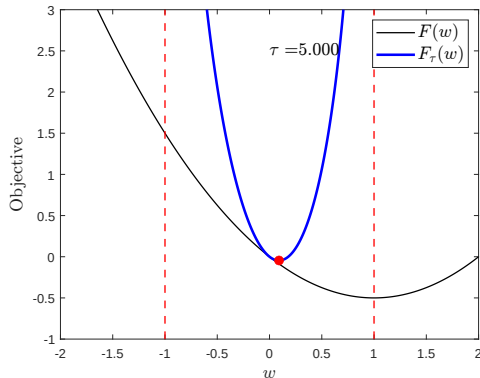
$$\chi(H_i(w)) = \begin{cases} 0 & \text{if } H_i(w) \geq 0 \\ \infty & \text{if } H_i(w) < 0 \end{cases}$$

Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

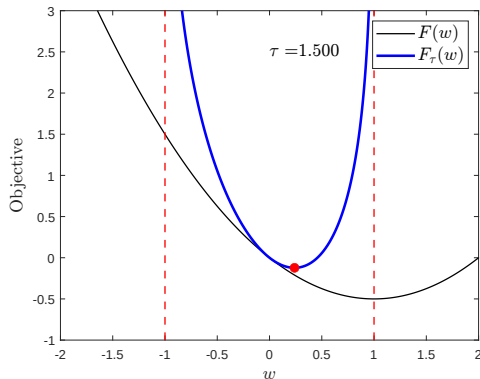


Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

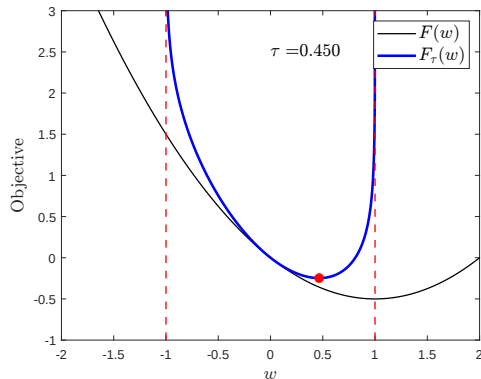


Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

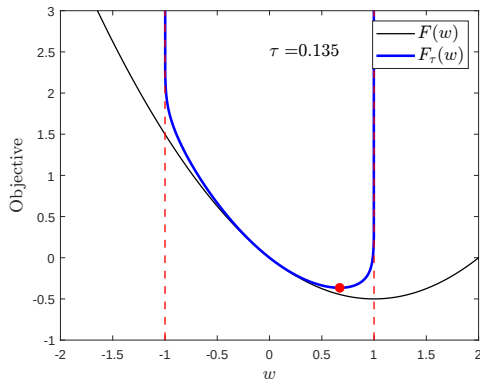


Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

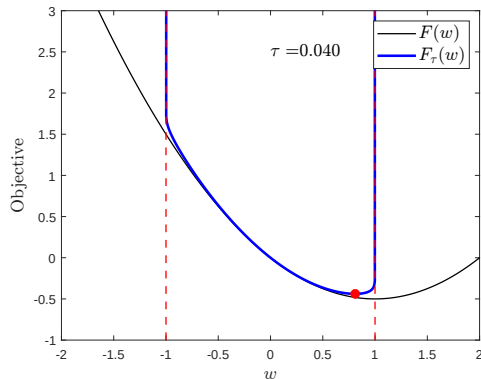


Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

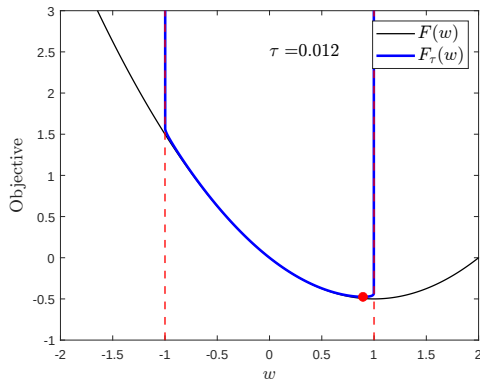


Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$

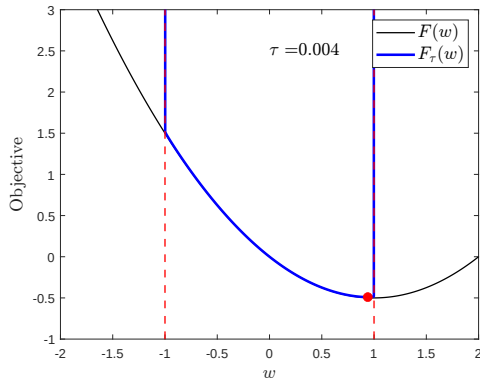


Example NLP

$$\begin{aligned} \min_w \quad & 0.5w^2 - 2w \\ \text{s.t.} \quad & -1 \leq w \leq 1 \end{aligned}$$

Barrier problem

$$\min_w 0.5w^2 - 2 - \tau \log(w + 1) - \tau \log(1 - w)$$



If one aims to solve a barrier NLP with non-zero $\tau_{\min} > 0$, one can sequentially solve "barrier quadratic programming problems" which each finish at the same positive $\tau_{\min}$.

Barrier QP

$$\begin{aligned} \min_{\Delta w, s} \quad & \nabla F(w^k)^\top \Delta w + \frac{1}{2} \Delta w^\top A^k \Delta w - \tau_{\min} \sum_{i=1}^p \log(s_i) \\ \text{s.t.} \quad & \begin{cases} G(w^k) + \nabla G(w^k)^\top \Delta w &= 0 \\ H(w^k) + \nabla H(w^k)^\top \Delta w - s &= 0 \end{cases} \end{aligned}$$

with a Hessian approximation e.g.

$$A_{\text{exact}}^k = \nabla_w^2 \mathcal{L}(w^k, \lambda^k, \mu^k) \quad \text{or} \quad A_{\text{Gauss-Newton}}^k = \nabla R(w) \nabla R(w)^\top$$

Its solution Δw^k , λ_{BQP}^+ , μ_{BQP}^+ delivers the next iterate

$$w^{k+1} = w^k + \Delta w^k, \quad \lambda^{k+1} = \lambda_{\text{BQP}}^+, \quad \mu^{k+1} = \mu_{\text{BQP}}^+.$$

Constrained-Gauss Newton (CGN) and Sequential Barrier Quadratic Programming (SBQP) are part of the family of Sequential Convex Programming (SCP) methods [Griffin and Stewart 1961, Tran-Dinh and Diehl 2010, Messerer et al. 2021], which address NLP with convex structure.

NLP with Outer Convex Structure

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \phi(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & w \in \Omega \end{aligned}$$

Here, ϕ and Ω are convex, and G is nonlinear and differentiable.

Constrained-Gauss Newton (CGN) and Sequential Barrier Quadratic Programming (SBQP) are part of the family of Sequential Convex Programming (SCP) methods [Griffin and Stewart 1961, Tran-Dinh and Diehl 2010, Messerer et al. 2021], which address NLP with convex structure.

NLP with Outer Convex Structure

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \phi(w) \\ \text{s.t.} \quad & G(w) = 0 \\ & w \in \Omega \end{aligned}$$

Here, ϕ and Ω are convex, and G is nonlinear and differentiable.

Convex SCP Subproblem at Iterate w^k

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \phi(w) + (1/2)(w - w^k)^\top A^k (w - w^k) \\ \text{s.t.} \quad & G(w^k) + \nabla G(w^k)^\top (w - w^k) = 0 \\ & w \in \Omega \end{aligned}$$

With affine equalities, this is a convex problem. Its solution gives the next iterate w^{k+1} . The Hessian approximation A^k is often set to zero, and linear convergence can still be shown.

- ▶ Newton-type methods linearize the nonlinear functions in each iteration
- ▶ for inequalities, KKT conditions are non-smooth, but one can apply a generalized Newton's method, Sequential Quadratic Programming (SQP), which needs a QP solver
- ▶ QPs can be solved with interior point methods (e.g. HPIPM)
- ▶ one can also directly solve NLPs by interior point methods, bypassing the QP of SQP (e.g. IPOPT, FATROP)
- ▶ Sequential Convex Programming (SCP) methods keep all convex structure in the subproblems (which can be solved by powerful convex solvers), with examples Constrained Gauss-Newton (CGN) and Sequential Barrier Quadratic Programming (SBQP)

Parametric NLP in MPC

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \phi(w) \\ \text{s.t.} \quad & G(p, w) = 0 \\ & w \in \Omega \end{aligned}$$

Here, the parameter p enters G linearly. The aim is to solve a sequence of problems with parameters $\dots, p^k, p^{k+1}, \dots$

Parametric NLP in MPC

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \phi(w) \\ \text{s.t.} \quad & G(p, w) = 0 \\ & w \in \Omega \end{aligned}$$

Here, the parameter p enters G linearly. The aim is to solve a sequence of problems with parameters $\dots, p^k, p^{k+1}, \dots$

RTI subproblem for p^{k+1} given w^k

$$\begin{aligned} w^{k+1} = \arg \min_{w \in \mathbb{R}^n} \quad & \phi(w) + (1/2)(w - w^k)^\top A^k (w - w^k) \\ \text{s.t.} \quad & G(p^{k+1}, w^k) + \nabla G(w^k)^\top (w - w^k) = 0 \\ & w \in \Omega \end{aligned}$$