Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 60-1 (2026) 61–66

GP Warm-starting of Dual Decomposition – Learning How to Speed Up ⋆

Rahul Panicker * **and Timm Faulwasser** *

* *Institute of Control Systems, Hamburg University of Technology, 21079 Hamburg, Germany*

(email: rahul.panicker@tuhh.de, timm.faulwasser@ieee.org)

Abstract: Dual decomposition is a well-established algorithm to solve strongly convex optimization problems in decentralized fashion. However, as dual decomposition is a first-order method, it often takes a significant number of iterations to converge to acceptable accuracy. We propose a novel dual warm-starting strategy based on learning a predictor of the optimal dual variable of the consensus constraints. We use Gaussian process regression to learn the predictor from pre-computed data. Furthermore, we exploit the posterior variance error bounds of Gaussian Processes and draw upon numerical examples to illustrate the advantages of our approach.

Copyright © 2026 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: Distributed optimization, warm-starting, gaussian processes, machine learning, dual decomposition, posterior covariance, error bound

1. INTRODUCTION

Distributed and decentralized optimization are of increasing interest for the learning and control of dynamic systems and beyond. Examples range from federated learning via the Alternating Direction Method of Multipliers (ADMM) (Zhou et al., 2023) to distributed Model Predictive Control (dMPC) (Camponogara et al., 2002).

It is worth noting that distributed optimization algorithms have a long history. The seminal work of Hestenes (1969) introduced the method of multipliers to solve equality constrained problems. Building on these ideas, Glowinski et al. (1975) and Gabay et al. (1976) proposed the Alternating Direction Method of Multipliers (ADMM). Dual Decomposition (DD), a gradient-based method to solve the dual problem, was introduced by Uzawa (1958) and later extended to the non-convex setting by Everett III (1963). These algorithms have found renewed interest across a range of applications, from control to learning (Luis et al., 2020; Stomberg, Schwan, et al., 2025).

The role of numerical optimization for Machine Learning (ML) can hardly be overstated (Sra et al., 2011; Sun et al., 2019). In contrast, using ML to improve optimization algorithms is an area of increasing research interest, aiming to improve the speed of convergence among other aspects. For example, Gregor et al. (2010) attempts to learn the argmin operator using a nonlinear predictor, enabling faster convergence.

Likewise, distributed optimization algorithms can benefit from tailored integration with ML. Examples include reinforcement learning for parameter selection in distributed power flow problems (Zeng et al., 2022) and deep learning-based acceleration of ADMM (Doerks et al., 2025), both of which target improved convergence speed.

Beyond parameter tuning, ML-based acceleration also applies to improved warm-starting. It is known in numerical optimization that the runtime of an algorithm depends on the quality of the initial guesses. ADMM, e.g., requires initial guesses for both the primal and dual variables, whereas DD only requires a dual initialization. Chanfreut et al. (2021) demonstrate that a neural network-based dual predictor can be used to warm-start the DD algorithm, leading to faster convergence.

In this paper, we use Gaussian Process (GP) regression to warm-start the DD algorithm. In contrast to Chanfreut et al. (2021), GP regression allows for formal error analysis by linking the error of the learned predictor to the algorithm performance. GP regression estimates the relation between the parameters of a strongly convex, affinely coupled, partially separated multi-parametric Nonlinear Program (NLP) and the dual variables corresponding to its coupling (consensus) constraints.

Furthermore, we utilize the posterior covariance of the learned GP to provide an upper-bound on the primal suboptimality and the number of DD iterations necessary to obtain it. We also propose a GP regression based warm-starting algorithm that is tested on a numerical example. The results demonstrate a 65% reduction in the number of iterations taken for the algorithm to converge.

The rest of this paper is organized as follows: In Section 2, we introduce the problem setting. Section 3 details the aforementioned error bound, Section 4 the numerical example, and finally Section 5 concludes the paper.

⋆ The authors acknowledge support by the Deutsche Forschungsgemeinschaft – DFG (project number 527447339).

2405-8963 Copyright © 2026 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2026.09.011

2. PROBLEM FORMULATION

Consider a network of S agents, connected by a bidirectional communication graph $\mathcal{G} = (\mathcal{S}, \mathrm{E})$, where $\mathcal{S} = \{1, \dots, S\}$, and the edges $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{S}$ couple the agents. The set of neighboring agents of agent i is given by

$$\mathcal{N}_i = \{j \in \mathcal{S} \setminus \{i\} \mid (i,j) \in \mathcal{E}\}.$$

These agents cooperatively solve the following centrally formulated, strongly convex, multi-parametric Quadratic Program (mp-QP)

$$\min_{\mathbf{x}} \quad \frac{1}{2}\,\mathbf{x}^\top \mathbf{H}\mathbf{x} + \mathbf{f}^\top(p)\mathbf{x} \tag{1a}$$
$$\text{subject to} \quad A_{eq}\mathbf{x} = b_{eq}, \tag{1b}$$
$$A_{in}\mathbf{x} \le b_{ineq}, \tag{1c}$$

parameterized by $p \in \mathbb{P} \subset \mathbb{R}^{n_p}$. The vector $\mathbf{x} = [x_1^\top, \dots, x_S^\top]^\top \in \mathbb{R}^n$ contains variables $x_i \in \mathbb{R}^{n_{x_i}}$ local to each agent $i \in \mathcal{S}$.

Problem (1) describes a coupled mp-QP as the matrix $\mathbf{H}$ contains off-diagonal terms, and the matrices A_{eq}, A_{in} couple neighboring agent variables. To solve this problem in a distributed fashion, we transform it into an affinely coupled, partially separable mp-QP (Engelmann et al., 2017; Stomberg, Engelmann, et al., 2022). To this end, the coupled constraints (1b) and (1c) are partitioned into local equalities and inequalities.

For a single coupled equality constraint, we have

$$a_{ii}^\top x_i + \sum_{j \in \mathcal{N}_i} a_{ij}^\top x_j = b_i^{eq}.$$

We introduce the variable v_{ij} as a copy of x_j available to agent i. This transforms the constraint into

$$a_{ii}^\top x_i + \sum_{j \in \mathcal{N}_i} a_{ij}^\top v_{ij} = b_i^{eq}.$$

The new decision variable of agent i is

$$z_i := [x_i, \boldsymbol{v}_i] \in \mathbb{R}^{n_i},$$

where $x_i \in \mathbb{R}^{n_{x_i}}$ are the variables of agent i, and $\boldsymbol{v}_i = [v_{ij}]_{j \in \mathcal{N}_i} \in \mathbb{R}^{n_{v_i}}$ collects copies of the variables of all neighbors of agent i. The partitioned constraints now read $A_i^{\mathcal{E}} z_i = b_i^{\mathcal{E}}$ and $A_i^{\mathcal{I}} z_i \le b_i^{\mathcal{I}}$ $\forall\, i \in \mathcal{S}$. Observe that the constraints are now decoupled since the variable z_i is local to agent i.

The cost is treated in a similar fashion, where each instance of the neighboring agent's variable is replaced with the copy v_{ij}, and from the new decision variable z_i we obtain

$$\phi_i(z_i, p) = \frac{1}{2}\, z_i^\top H_i z_i + f_i^\top(p) z_i + \iota_{\mathbb{Z}_i}.$$

For notational simplicity we move the local constraints to the cost via the indicator function $\iota_{\mathbb{Z}_i}$, where $\mathbb{Z}_i := \{z_i \mid A_i^{\mathcal{E}} z_i = b_i^{\mathcal{E}},\ \ A_i^{\mathcal{I}} z_i \le b_i^{\mathcal{I}}\}$ is the feasible set of agent i.

To ensure consensus among the variable copies and the originals we formulate global affine coupling constraints, $v_{ij} - x_j = 0$, in matrix-vector form as (2b). The matrix $E_i \in \mathbb{R}^{n_c \times n_i}$ is a block matrix of appropriate dimension representing n_c coupling constraints. The Lagrange multipliers corresponding to the consensus constraints are given by $\lambda \in \mathbb{R}^{n_c}$, and are termed the *consensus duals*. The affinely coupled form of the mp-QP is

Algorithm 1: Accelerated Dual Decomposition

Inputs: $p,\ \lambda^0,\ k_{\max}$
Initialize: $L = \|EH^{-1}E^\top\|_2\ ,\ \ \nu^0 = \lambda^0\ ,\ \lambda^{-1} = \lambda^0$
while $k < k_{\max}$ ***and*** $\sum_{i \in \mathcal{S}} E_i z_i^k \neq 0$ **do**
  $\beta = \frac{k-1}{k+2}$
  $\nu^k = \lambda^k + \beta(\lambda^k - \lambda^{k-1})$
  $z_i^k = \underset{z_i}{\operatorname{argmin}}\, \phi_i(z_i, p) + \nu^{k\top} E_i z_i \quad \forall i \in \mathcal{S}$
  $\lambda^{k+1} = \nu^k + \frac{1}{L}\left(\sum_{i \in \mathcal{S}} E_i z_i^k\right)$
return λ^k, z_i^k

$$\min_{\{z_i\}_{i \in \mathcal{S}}} \quad \sum_{i \in \mathcal{S}} \phi_i(z_i, p) \tag{2a}$$
$$\text{subject to} \quad \sum_{i \in \mathcal{S}} E_i z_i = b \quad \mid \quad \lambda. \tag{2b}$$

2.1 Dual Decomposition

Dual decomposition (DD) is a decentralized [1] optimization algorithm for solving convex QPs based on duality theory (Bertsekas, 1999; Boyd et al., 2010). To this end, we first define the dual function of (2), as

$$q(\lambda) = \inf_{\{z_i\}_{i \in \mathcal{S}}} \mathcal{L}(z, \lambda; p),$$

where $\mathcal{L}$ is the Lagrangian, given by

$$\mathcal{L}(z, \lambda; p) = \sum_{i \in \mathcal{S}} \phi_i(z_i, p) + \lambda^\top (E_i z_i) - \lambda^\top b.$$

The dual problem is then defined as

$$\sup_{\lambda} q(\lambda).$$

The idea behind the DD algorithm is using gradient ascent to solve the dual problem. Under *Slater's condition*, (2) exhibits no duality gap (Bertsekas, 1999). The DD iterations are given by

$$z_i^{k+1} = \underset{z_i}{\operatorname{argmin}}\, \phi_i(z_i, p) + \lambda^{k\top} E_i z_i, \quad \forall\, i \in \mathcal{S},$$
$$\lambda^{k+1} = \lambda^k + \alpha \left(\sum_{i \in \mathcal{S}} E_i z_i^{k+1} \right).$$

Bertsekas (1999, Prop. 1.2.3) shows that convergence is guaranteed for any step size

$$\alpha \in \left(0, \frac{2\mu}{\|\mathbf{E}\|_2^2}\right),\ \mathbf{E} = [E_1, \dots, E_S],$$

where μ is the smallest eigenvalue of the Hessian of $\mathcal{L}$.

DD is attractive due to its low implementation complexity. However, like other classical gradient-based methods, it is slow to converge. The standard DD algorithm has a sublinear convergence of $\mathcal{O}(1/k)$ (Beck et al., 2014).

There have been efforts to improve this convergence rate. Conte et al. (2012) and Giselsson et al. (2013) implement Nesterov's accelerated gradient method to increase the speed of convergence to sublinear $\mathcal{O}(1/k^2)$. Algorithm 1

[1] Decentralized algorithms only require communication between subsystems. Distributed algorithms require communication between subsystems, as well as a limited amount of central coordination (Bertsekas and Tsitsiklis, 1989).

details the accelerated gradients implementation by Giselsson et al. (2013). Observe that only dual information is necessary for initialization.

Lemma 1. (Giselsson et al. (2013)). Consider the application of Algorithm 1 to strongly convex QPs. Then it holds that

$$\|\mathbf{z}^k - \mathbf{z}^\star\|_2^2 \;\leq\; \mathrm{C}\frac{\|\lambda^0 - \lambda^\star\|_2^2}{(k+1)^2}, \tag{4}$$

where $\mathbf{z}^k = [z_1^{k\top}, \ldots, z_S^{k\top}]^\top$ is the k-th primal iterate, $\lambda^\star$ is the optimal dual value, and $\mathrm{C} \in \mathbb{R}^+$ is a constant.

Corollary 2. (One step convergence). If $\lambda^0 = \lambda^\star$, then the dual value, and by extension the primal value, will converge to the optimum in $k = 1$ iteration. □

It is known for an mp-QP that the dual variable is a function of the parameters (Bemporad et al., 2002, Thm. 2). This observation is our core motivation to learn a predictor that maps p to the dual variables of the consensus constraints

$$\Delta : \mathbb{R}^{n_p} \supset \mathbb{P} \ni p \mapsto \lambda \in \mathbb{R}^{n_c} \tag{5}$$

where $\mathbb{P}$ is an open subset of $\mathbb{R}^{n_p}$.

2.2 Gaussian Process Regression

Gaussian Process (GP) regression is a non-parametric regression method for approximating nonlinear functions from data (Rasmussen et al., 2006). A GP describes a Gaussian distribution over functions, whereby unknown functions estimated using GPs are fully characterized by their mean and covariance functions. The prior distribution encodes any existing knowledge about the function. We assume a zero mean Gaussian prior and define the prior covariance by the *kernel function* κ.

Consider N uniquely sampled parameters, $p_i \in \bar{\mathbb{P}},\ i = \{1, \ldots, N\}$ and their corresponding optimal dual variables $\lambda_i \in \mathbb{R}^{n_c}$. We use $\mathcal{D}_p := [p_1^\top, \ldots, p_N^\top] \in \mathbb{R}^{N \times n_p}$ to denote the collected inputs, and $\mathcal{D}_\lambda := [\lambda_1^\top, \ldots, \lambda_N^\top] \in \mathbb{R}^{N \times n_c}$ for the outputs. The training dataset $\mathcal{D} := \{(p_i, \lambda_i)\}_{i \in \mathbb{N}_{[1,N]}} \subset \bar{\mathbb{P}} \times \mathbb{R}^{n_c}$, whereby $\bar{\mathbb{P}}$ is a suitable compact subset of $\mathbb{P}$.

GP regression establishes the conditional relationship for a new point $p \in \bar{\mathbb{P}}$

$$\lambda \mid p, \mathcal{D} \sim \mathcal{N}\big(\hat{\Delta}, \sigma^2\big),$$

where the posterior mean and variance functions are

$$\hat{\Delta}(p) = \mathbf{k}(\cdot, p)^\top \left(\mathbf{K} + \sigma_n^2 I\right)^{-1} \mathcal{D}_\lambda, \tag{6}$$

$$\sigma^2(p) = \kappa(p, p) - \mathbf{k}(\cdot, p)^\top \left(\mathbf{K} + \sigma_n^2 I\right)^{-1} \mathbf{k}(\cdot, p). \tag{7}$$

The prior covariance matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ (or Gramian) is defined as

$$\mathbf{K}_{ij} = \kappa(p_i, p_j),\ \forall\ i, j \in \mathbb{N}_{[1,N]}.$$

The column vector $\mathbf{k}(\cdot, p) = [\kappa(p_1, p), \ldots, \kappa(p_N, p)]^\top \in \mathbb{R}^N$ is the evaluation of the kernel function of the new point p at each sampled input point.

The choice of the kernel function κ determines the Reproducing Kernel Hilbert Space (RKHS) for function estimation (Berlinet et al., 2011, Theorem 3). The parameters of the kernel, along with an additional process noise covariance parameter, are called hyperparameters $\theta \in \mathbb{R}^{n_\theta + 1}$. The matrix $\mathbf{K}$ and the vector $\mathbf{k}$ depend on these hyperparameters and on the input. We therefore write them as $\mathbf{K}(\theta, \mathcal{D}_p)$ and $\mathbf{k}(\theta, \mathcal{D}_p)$ (Ferreira et al., 2018).

We consider $\mathbf{M} := \mathbf{K}(\theta, \mathcal{D}_p) + \sigma_n^2 I$ where I is the identity matrix of appropriate dimension and σ_n^2 is the process noise covariance that regularizes the $\mathbf{K}$ matrix, cf (6). The hyperparameters θ are obtained by solving

$$\theta^\star = \underset{\theta}{\operatorname{argmax}}\ \Pi(\theta, \mathcal{D}), \tag{8}$$

where $\Pi(\theta, \mathcal{D})$ is the log-marginal likelihood

$$\Pi(\theta, \mathcal{D}) = -\frac{1}{2}\mathcal{D}_\lambda^\top \mathbf{M}^{-1} \mathcal{D}_\lambda - \frac{1}{2}\log|\mathbf{M}| - \frac{N}{2}\log 2\pi.$$

Lemma 3. (De Marchi et al., 2011; Molodchyk et al., 2024) If the true function $\Delta \in \mathcal{H}$, where $\mathcal{H}$ is the RKHS induced by the kernel κ, then the upper bound on the approximation error in relation to the posterior variance of the GP is given by

$$\|\Delta(p) - \hat{\Delta}(p)\|_2^2 \;\leq\; \sigma^2(p) \sum_{j=1}^{n_c} \|\Delta_j\|_{\mathcal{H}}^2, \tag{9}$$

where $\|\cdot\|_{\mathcal{H}}$ is the RKHS norm of $\mathcal{H}$.

Remark 4. Lemma 3 as stated is valid for noise-free interpolation. The training targets in this work are synthetic and are therefore essentially noise-free. The process noise covariance σ_n^2 is used purely as numerical regularization. The bound (9) remains valid for the regularized predictor in (6) with $\sigma^2(p)$ as in (7) (Maddalena et al., 2021).

3. GP ACCELERATED DUAL WARM-STARTING

Motivated by Lemma 1 and Corollary 2, we propose to employ GP regression to estimate a dual predictor for warm-starting DD.

GP regression is chosen in this paper for the following reasons: (i) certain kernels are universal, i.e., their RKHS can approximate any continuous function on a compact

Algorithm 2: GP warm-started dual decomposition

Offline – GP regression

Collect training data $\mathcal{D}$

$\theta^\star \leftarrow$ Solve optimization problem (8)

$\hat{\Delta}(\cdot) \leftarrow$ Compute function (6) using $\theta^\star$

return $\hat{\Delta}(\cdot)$

Online – Warm-started dual decomposition

Inputs: $p,\ k_{max},\ \hat{\Delta}(\cdot),\ L = \|EH^{-1}E^\top\|_2$

Initialize: $\lambda^0 \leftarrow \hat{\Delta}(p),\ \nu^0 = \lambda^0,\ \lambda^{-1} = \lambda^0$

while $k < k_{\max}$ ***and*** $\sum_{i \in \mathcal{S}} E_i z_i^k \neq b$ **do**

$\beta = \frac{k-1}{k+2}$

$\nu^k = \lambda^k + \beta(\lambda^k - \lambda^{k-1})$

$z_i^k = \underset{z_i}{\operatorname{argmin}}\, \phi_i(z_i, p) + \nu^{k\top} E_i z_i \quad \forall\ i \in \mathcal{S}$

$\lambda^{k+1} = \nu^k + \frac{1}{L}\left(\sum_{i \in \mathcal{S}} E_i z_i^k\right)$

return λ^k, z_i^k

domain (Micchelli et al., 2006); and (ii) it is possible to represent the approximation error bound as a function of the posterior variance (cf. Lemma 3).

However, Lemma 3 requires $\Delta \in \mathcal{H}$ for a known kernel κ that is used during the training process. Since Δ from (5) is piecewise affine (Bemporad et al., 2002), it generally does not belong to the RKHS associated with kernels commonly used for GP regression.

For Matérn kernels, RKHS membership can be established but only under restrictive assumptions. Sobolev spaces $\mathrm{W}^{\nu+\frac{n_p}{2},2} = \mathrm{H}^{\nu+\frac{n_p}{2}}$ form the RKHS of the Matérn kernel with smoothness ν (Kanagawa et al., 2018). A generic piecewise affine function is contained in the RKHS for sufficiently small $\nu + \frac{n_p}{2}$. For the roughest Matérn kernel ($\nu = \frac{1}{2}$) this restricts the dimension to $n_p = 1$(Di Nezza et al., 2012).

This motivates reformulation of the parametic QP (2) with a smoother alternative. Adding the inequality constraints to the cost function via a log-barrier function yields

$$\min_{\{z_i\}_{i\in\mathcal{S}}} \frac{1}{2}\sum_{i\in\mathcal{S}} z_i^\top H_i z_i + f_i^\top(p) z_i - \mu \sum_{\ell=1}^{n_i^{\mathcal{I}}} \ln(b_{i\ell}^{\mathcal{I}} - a_{i\ell}^{\mathcal{I}^\top} z_i) \quad (10a)$$

$$\text{subject to} \quad A_i^{\mathcal{E}} z_i = b_i^{\mathcal{E}},\ \forall i \in \mathcal{S} \quad (10b)$$

$$\sum_{i\in\mathcal{S}} E_i z_i - b = 0 \mid \lambda. \quad (10c)$$

Assumption 5. The objective function (10a) is strongly convex, $H_i \succ 0\ \forall\ i \in \mathcal{S}$.

Assumption 6. The domain of the map of optimal duals $\mathbb{P} \subset \mathbb{R}^{n_p}$ is open, and for every $p \in \mathbb{P}$ there exists z satisfying $A^{\mathcal{E}} z = b^{\mathcal{E}}$, $A^{\mathcal{I}} z < b^{\mathcal{I}}$.

Assumption 7. The block matrix $B^\top = [A^{\mathcal{E}\top}\ E^\top]$ where $A^{\mathcal{E}} = \mathrm{diag}(A_i^{\mathcal{E}})$, and $E = [E_1, \ldots, E_S]$ has full row rank.

Assumption 8. $f_i(\cdot)$ is affine in the parameter p.

Proposition 9. Suppose Assumptions 5–8 hold and consider (10). Then there exists a unique function $\Delta_\mu : \mathbb{P} \ni p \mapsto \lambda_\mu \in \mathbb{R}^{n_c}$ and $\Delta_\mu \in C^\infty(\mathbb{P}, \mathbb{R}^{n_c})$.

Proof. The Lagrangian of (10) is given by

$$\mathcal{L} = \frac{1}{2}\sum_{i\in\mathcal{S}} z_i^\top H_i z_i + f_i^\top(p) z_i + \gamma_i^\top \left(A_i^{\mathcal{E}} z_i - b_i^{\mathcal{E}}\right) - \mu \sum_{\ell=1}^{n_i^{\mathcal{I}}} \ln(s_{i\ell}) + \lambda^\top \left(\sum_{i\in\mathcal{S}} E_i z_i - b\right)$$

where $s_{i\ell} = b_{i\ell}^{\mathcal{I}} - a_{i\ell}^{\mathcal{I}^\top} z_i$. Stationarity of the Lagrangian is given as

$$\nabla_{z_i}\mathcal{L}_i = H_i z_i + f_i(p) + \mu A_i^{\mathcal{I}\top} M_i(z_i)^{-1}\mathbf{1} + A_i^{\mathcal{E}\top}\gamma_i + E_i^\top \lambda = 0 \quad (11a)$$

where $M_i(z_i) := \mathrm{diag}(s_{i1}, \ldots, s_{in_i^{\mathcal{I}}})$, and primal feasibility is given bys

$$A_i^{\mathcal{E}} z_i - b_i^{\mathcal{E}} = 0 \quad (11b)$$

$$\sum_{i\in\mathcal{S}} E_i z_i - b = 0. \quad (11c)$$

Stacking (11) for all agents $i \in \mathcal{S}$ and introducing the variable $w = [z^\top\ \gamma^\top\ \lambda^\top]^\top$ we obtain $\mathcal{F}(w, p) = 0$ where

$$\mathcal{F}(w,p) = \begin{bmatrix} Hz + f(p) + \mu A^{\mathcal{I}\top} M^{-1}\mathbf{1} + A^{\mathcal{E}\top}\gamma + E^\top\lambda \\ A^{\mathcal{E}} z - b^{\mathcal{E}} \\ Ez - b \end{bmatrix}. \quad (12)$$

The matrices $H, A^{\mathcal{I}}, A^{\mathcal{E}}, M$ are block diagonal, comprised of the per-agent matrices, $f(p) = [f_1^\top(p), \ldots, f_S^\top(p)]^\top$, and $E = [E_1\ \ldots\ E_S]$ with $\mathcal{F}$ defined on the open set $\{s_{i\ell} > 0\ \forall i \in \mathcal{S},\ \ell \in \mathbb{N}_{[1,n_i^{\mathcal{I}}]}\}$.

Assumptions 5–7 guarantee the existence of a unique solution $w^\star(p) = [z_\mu^{\star\top}\ \gamma^{\star\top}\ \lambda_\mu^{\star\top}]^\top$ for each $p \in \mathbb{P}$ interior to $\{s_{i\ell} > 0 \mid i \in \mathcal{S},\ \ell \in \mathbb{N}_{[1,n_i^{\mathcal{I}}]}\}$.

The Jacobian of (12), is given by the matrix

$$\frac{\partial \mathcal{F}(w,p)}{\partial w} = \begin{bmatrix} H_\mu & B^\top \\ B & 0 \end{bmatrix}$$

where $H_\mu = H + \mu A^{\mathcal{I}^\top} M^{-2} A^{\mathcal{I}} \succeq H$. This Jacobian is non-singular for any point $(w^\star(p), p)$ and all $p \in \mathbb{P}$ by Assumptions 5 and 7 (Benzi et al., 2005). Then, by the Implicit Function Theorem, $w^\star \in C^\infty(\mathbb{P}, \mathbb{R}^{n_c})$ and its λ-components are $\Delta_\mu \in C^\infty(\mathbb{P}, \mathbb{R}^{n_c})$ (Nocedal et al., 2006). ■

In practice, the input parameters are sampled from a compact $\bar{\mathbb{P}} \subset \mathbb{P}$. Therefore, by Proposition 9, Lemma 3 can be used to specify the error bound (9) for Δ_μ for kernels such as the Matérn-ν kernel.

Remark 10. Lemma 1 extends to (10) as its proof only requires strong convexity of the primal objective and Lipschitz continuity of the dual gradient, both of which hold for (10).

The proposed scheme for GP-based warm starting, detailed in Algorithm 2, contains an offline training phase where the dual predictor is learned from data. The online phase evaluates the pre-computed function for the current parameter value and uses the predicted dual as a warm-start for DD.

Proposition 11. Let $k \in \mathbb{N}$ be the iteration index of Algorithm 2, then the primal error at iteration k is bounded from above by

$$\|\mathbf{z}^k - \mathbf{z}_\mu^\star\|_2^2 \ \le\ \frac{\mathrm{C}}{(k+1)^2}\ \sigma^2(p)\ \sum_{j=1}^{n_c} \|\Delta_{\mu,j}\|_{\mathcal{H}}^2, \quad (13)$$

for $\mathrm{C} \in \mathbb{R}^+$.

Proof. Restating (4) for the initial guess $\lambda^0 = \hat{\Delta}(p)$ and the optimal dual $\lambda_\mu^\star = \Delta_\mu(p)$ for $p \in \mathbb{P}$ yields,

$$\|\mathbf{z}^k - \mathbf{z}_\mu^\star\|_2^2 \ \le\ \mathrm{C}\frac{\|\Delta_\mu(p) - \hat{\Delta}(p)\|_2^2}{(k+1)^2},$$

$$\|\Delta_\mu(p) - \hat{\Delta}(p)\|_2^2 \ \ge\ \frac{(k+1)^2}{\mathrm{C}}\|\mathbf{z}^k - \mathbf{z}_\mu^\star\|_2^2. \quad (14)$$

The constant C is given by

$$\mathrm{C} = \frac{4L}{\sigma_m(H)} \in \mathbb{R}^+.$$

Here, $\sigma_m(H)$ is the smallest eigenvalue of the block Hessian matrix, and $L = \|EH^{-1}E^\top\|_2$ is the smallest Lipschitz constant of the gradient of the dual function.

From (9) and (14), we get

$$\frac{(k+1)^2}{\mathrm{C}}\|\mathbf{z}^k - \mathbf{z}_\mu^\star\|_2^2 \leq \sigma^2(p)\sum_{j=1}^{n_c}\|\Delta_{\mu,j}\|_{\mathcal{H}}^2,$$

where the variance $\sigma^2(p)$ is computed as per (7). ■

Proposition 12. (Iteration bound for maximum error). For any $\varepsilon > 0$, if Algorithm 2 is applied for at least $\hat{k}$ iterations, with

$$\hat{k} = \sqrt{\frac{\mathrm{C}}{\varepsilon^2}\sigma^2(p)\sum_{j=1}^{n_c}\|\Delta_{\mu,j}\|_{\mathcal{H}}^2} - 1,$$

then $\|\mathbf{z}^{\hat{k}} - \mathbf{z}_\mu^\star\|_2 \leq \varepsilon$ holds.

Proof. The maximum error occurs when (13) holds with equality, i.e.,

$$\varepsilon^2 = \frac{\mathrm{C}}{(\hat{k}+1)^2}\sigma^2(p)\sum_{j=1}^{n_c}\|\Delta_{\mu,j}\|_{\mathcal{H}}^2.$$

Rearranging in terms of $\hat{k}$ proves the result. ■

4. NUMERICAL EXAMPLE

We consider a four node sensor network that collaboratively estimates the parameter $\theta \in \mathbb{R}^2$. The sensors are connected via a bidirectional, ideal path graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$. All sensors have the linear measurement equation given by

$$y_i = M\theta + d,\ M = \begin{bmatrix} 1 & 0.1 \\ 0.1 & 2 \end{bmatrix}$$

where $d \sim \mathcal{N}(0,1)$ is the measurement noise.

The estimation problem is formulated as a distributed mp-QP by introducing local copies of the variable θ for each agent and enforcing consensus between these copies

$$\min_{\{\theta_i\}_{i\in\mathcal{S}}} \frac{1}{2}\sum_{i=1}^{4}\|M\theta_i - y_i\|_2^2, \tag{15a}$$

$$\text{subject to}\ \ \theta_i - \theta_j = 0 \mid \lambda\ , \forall (i,j) \in \mathcal{E}. \tag{15b}$$

Here, θ_i is the local copy of the variable to be estimated and y_i is the measurement of each sensor, and the parameter of this mp-QP. This is the simplest case of (10), where there are no inequality constraints.

Training targets are collected by solving (15) centrally using `IPOPT` (Wächter et al., 2006) for 5000 unique values of the parameter which are sampled using Latin Hypercube Sampling (LHS) (Loh, 1996), 85% of which are used as training data and the rest used as validation data. The models are trained using the `BoTorch` library in Python (Balandat et al., 2020) on a 10GB Nvidia GeForce RTX 3080 GPU. The source code is available online. [2]

Fig. 1 provides a statistical summary of the accuracy obtained with the Matérn kernel. The two box plots show the distributions of the ℓ_2 and ℓ_∞ norms of the dual prediction error evaluated on 375 random samples from the validation dataset.

Since (15) is a purely equality constrained mp-QP, the optimal dual mapping is affine. Consequently, a linear kernel can represent this mapping exactly up to machine

[2] `https://github.com/OptCon/GP-warmstart-DD`

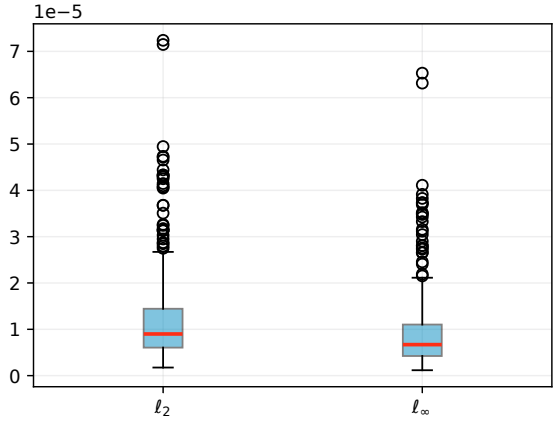


Fig. 1. 2-norm and ∞-norm of the validation error for the Matérn kernel

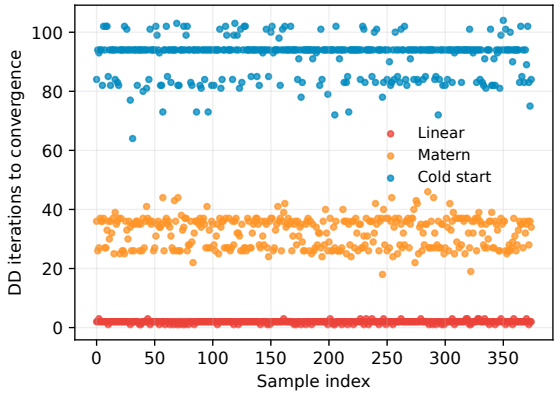


Fig. 2. DD iterations for warm-start vs cold-start

accuracy. This observation is corroborated by the convergence behavior of Algorithm 2 illustrated in Fig. 2. It compares the number of iterations required for the consensus residual to fall below the required tolerance of 10^{-9} for both the Matérn kernel and the Linear kernel. While the Matérn kernel improves convergence by around 65%, the Linear kernel causes Algorithm 2 to converge in a few steps, as predicted in Corollary 2.

The affine map of (15) is explicitly derivable owing to the structure of the problem; this means for the linear kernel one can compute the bounds (13) exactly. However, this is in general not the case.

5. CONCLUSIONS

In this paper, we propose a novel dual warm-starting methodology that learns a dual predictor using GP regression. The learned dual predictor is used to estimate the optimal dual variable, which in turn is used to warm-start the DD algorithm for a barrier smoothed mp-QP. Furthermore, we provide a formal error bound on the primal convergence of the warm-started algorithm and prove that a smoothed piecewise affine function lies in the RKHS of the Matérn class of kernels.

The proposed algorithm is tested on a numerical example, where the performance of the warm-started DD algorithm is compared to a cold-start baseline for two kernel choices. The results show that the average reduction in the number

of DD iterations was roughly 65% while training using the Matérn kernel. The linear kernel predicts the optimal dual accurately to machine precision and acclerated convergence by almost two orders of magnitude compared to cold-started DD iterations. However, it was empirically observed that GP regression does not provide the approximation quality required for warm-starting problems of increased complexity. Additionally, tuning GP hyperparameters for large scale problems proved to be a bottleneck.

Future research will thus focus on predictors that scale well with the number of dual variables and it will also investigate how one can pre-process the optimization problem to reduce the computational cost of the learning process.

DECLARATION OF GENERATIVE AI

During the preparation of this work the author(s) used generative AI, in order to improve wording and grammar for certain sections.

REFERENCES

Balandat, M. et al. (2020). "BoTorch: A Framework for Efficient Monte-Carlo Bayesian Optimization". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 21524–21538.

Beck, A. et al. (2014). "An $\mathcal{O}(1/k)$ Gradient Method for Network Resource Allocation Problems". In: *IEEE Transactions on Control of Network Systems* 1.1, pp. 64–73.

Bemporad, A. et al. (2002). "The explicit linear quadratic regulator for constrained systems". In: *Automatica* 38.1, pp. 3–20.

Benzi, Michele et al. (2005). "Numerical solution of saddle point problems". In: *Acta Numerica* 14, pp. 1–137.

Berlinet, A. et al. (2011). *Reproducing kernel Hilbert spaces in probability and statistics*. Springer Science & Business Media.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Belmont, MA: Athena Scientific.

Bertsekas, D. P. and John N. Tsitsiklis (1989). *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ: Prentice-Hall.

Boyd, S. et al. (2010). "Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers". In: *Foundations and Trends in Machine Learning* 3.1, pp. 1–122.

Camponogara, E. et al. (2002). "Distributed model predictive control". In: *IEEE Control Systems Magazine* 22.1, pp. 44–52.

Chanfreut, P. et al. (2021). "Distributed Model Predictive Control based on Dual Decomposition with Neural-Network-based Warm Start". In: *2021 European Control Conference (ECC)*, pp. 1969–1974.

Conte, C. et al. (2012). "Computational aspects of distributed optimization in model predictive control". In: *Proceedings of the 51st IEEE Conference on Decision and Control (CDC)*, pp. 6819–6824.

De Marchi, S. et al. (2011). "Kernel functions and meshless methods". In: *Dolomites Research Notes on Approximation* 4, pp. 1–63.

Di Nezza, Eleonora et al. (2012). "Hitchhiker's guide to the fractional Sobolev spaces". In: *Bulletin des Sciences Mathématiques* 136.5, pp. 521–573.

Doerks, H. et al. (2025). "Learning to accelerate distributed ADMM using graph neural networks". In: *arXiv:2509.05288*.

Engelmann, A. et al. (2017). "Distributed AC Optimal Power Flow using ALADIN." In: *IFAC-PapersOnLine* 50.1, pp. 5536–5541.

Everett III, H. (1963). "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources". In: *Operations research* 11.3, pp. 399–417.

Ferreira, T. de Avila et al. (2018). "Real-Time optimization of Uncertain Process Systems via Modifier Adaptation and Gaussian Processes". In: *2018 European Control Conference (ECC)*, pp. 465–470.

Gabay, D. et al. (1976). "A dual algorithm for the solution of nonlinear variational problems via finite element approximation". In: *Computers & mathematics with applications* 2.1, pp. 17–40.

Giselsson, P. et al. (2013). "Accelerated gradient methods and dual decomposition in distributed model predictive control". In: *Automatica* 49.3, pp. 829–833.

Glowinski, R. et al. (1975). "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires". In: *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 9.R2, pp. 41–76.

Gregor, K. et al. (2010). "Learning fast approximations of sparse coding". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning*. ICML'10. Omnipress, pp. 399–406. ISBN: 9781605589077.

Hestenes, M. R. (1969). "Multiplier and gradient methods". In: *Journal of optimization theory and applications* 4.5, pp. 303–320.

Kanagawa, Motonobu et al. (2018). "Gaussian processes and kernel methods: A review on connections and equivalences". In: *arXiv:1807.02582*.

Loh, W. L. (1996). "On Latin hypercube sampling". In: *The Annals of Statistics* 24.5, pp. 2058–2080.

Luis, C. E. et al. (2020). "Online Trajectory Generation With Distributed Model Predictive Control for Multi-Robot Motion Planning". In: *IEEE Robotics and Automation Letters* 5.2, pp. 604–611.

Maddalena, Emilio Tanowe et al. (2021). "Deterministic error bounds for kernel-based learning techniques under bounded noise". In: *Automatica* 134, p. 109896.

Micchelli, C. A. et al. (2006). "Universal Kernels". In: *Journal of Machine Learning Research* 7, pp. 2651–2667.

Molodchyk, O. et al. (2024). "Exploring the Links Between the Fundamental Lemma and Kernel Regression". In: *IEEE Control Systems Letters* 8, pp. 2045–2050.

Nocedal, Jorge et al. (2006). *Numerical optimization*. Springer.

Rasmussen, C. E. et al. (2006). *Gaussian Processes for Machine Learning*. Cambridge, MA: MIT Press.

Sra, S. et al. (2011). *Optimization for machine learning*. MIT press.

Stomberg, G., A. Engelmann, et al. (2022). "A compendium of optimization algorithms for distributed linear-quadratic MPC". In: *at-Automatisierungstechnik*, pp. 317–330.

Stomberg, G., R. Schwan, et al. (2025). "Cooperative distributed model predictive control for embedded systems: Experiments with hovercraft formations". In: *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 11377–11383.

Sun, S. et al. (2019). "A survey of optimization methods from a machine learning perspective". In: *IEEE transactions on cybernetics* 50.8, pp. 3668–3681.

Uzawa, H. (1958). "Iterative methods for concave programming". In: *Studies in Linear and Nonlinear Programming* 6, pp. 154–165.

Wächter, Andreas et al. (Mar. 1, 2006). "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1, pp. 25–57.

Zeng, S. et al. (2022). "A reinforcement learning approach to parameter selection for distributed optimal power flow". In: *Electric Power Systems Research* 212, p. 108546.

Zhou, S. et al. (2023). "Federated Learning Via Inexact ADMM". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.8, pp. 9699–9708.