Joint Constrained Bayesian Optimization of Planning, Guidance, Control, and State Estimation of an Autonomous Underwater Vehicle[†]

David Stenger[1], Maximilian Nitsch[1] and Dirk Abel[1]

Abstract— The performance of a guidance, navigation and control (GNC) system of an autonomous underwater vehicle (AUV) heavily depends on the correct tuning of its parameters. Our objective is to automatically tune these parameters with respect to arbitrary high-level control objectives within different operational scenarios. In contrast to literature, an overall tuning is performed for the entire GNC system, which is new in the context of autonomous underwater vehicles. The main challenges in solving the optimization problem are computationally expensive objective function evaluations, crashing simulations due to infeasible parametrization and the numerous tunable parameters (in our case 13). These challenges are met by using constrained Bayesian optimization with crash constraints. The method is demonstrated in simulation on a GNC system of an underactuated miniature AUV designed within the TRIPLE-nanoAUV initiative for exploration of subglacial lakes. We quantify the substantial reduction in energy consumption achieved by tuning the overall system. Furthermore, different parametrizations are automatically generated for different power consumption functions, robustness, and accuracy requirements. E.g. energy consumption can be reduced by $\sim 28\%$, if the maximum allowed deviation from the planned path is increased by $\sim 65\%$. This shows the versatile practical applicability of the optimization-based tuning approach.

I. INTRODUCTION

Successful operation of an autonomous underwater vehicle (AUV) requires the close collaboration of state estimation, guidance, control, and path planning algorithms within a guidance navigation and control (GNC) framework. For each of those domains appropriate algorithms were established in simulation and real world experiments e.g. [1], [2], [3].

However, the performance of the overall system heavily depends on the correct tuning of their algorithmic parameters for the specific task and vehicle at hand. These parameters govern e.g. the aggressiveness of the guidance algorithm. For some GNC algorithms, analytical tuning methods exist. However, they can only be applied to a limited range of scenarios and control objectives (e.g. LQR). In contrast, we focus on the tuning of the overall GNC system and allow arbitrary high-level objective functions and constraints.

The tuning problem is formulated as a **black-box optimization problem** to be approximately solved in simulation. Solving black-box problems is challenging due to the unknown analytic relationship between the GNC parameters and the obtained objective function values. Additionally, poor

parametrization may result e.g. in diverging state estimation solutions leading to simulation crash. This setting is known as learning under crash constraints (LCC) [4]. Furthermore, closed-loop simulations are expensive in terms of CPU time. Thus, sample-efficiency of the used optimizer is important.

In this contribution, we use constrained **Bayesian optimization (BO)** with crash constraints to meet these challenges. BO has become a common method used for sample-efficient black-box optimization e.g. in control [5] and state estimation [6]. Constrained BO in control engineering [7] as well as LCC [4] have previously been addressed.

In the context of **automatic tuning for AUVs**, to the best of the authors knowledge, tuning of the overall GNC system has not been considered yet. Different metaheuristics have been applied for PID tuning for AUVs [8]. However, such metaheuristics were shown to be less sample-efficient than BO (cf. e.g. [9], [10]). BO has been used to fairly compare different controllers for AUVs [11]. Simplified optimal state knowledge was assumed, though. The tuning of parameters of navigation filters for AUVs has only been reported using partical swarm optimization [12]. E.g. in [6], the potential of BO for filter tuning has been show for a different application. However, in those publications the effect of filter design on the control performance was not considered.

In this contribution, we showcase the BO-based tuning method on an underactuated AUV designed for the exploration of subglacial lakes and oceans within the **TRIPLE-nanoAUV** initiative [13], [14]. For this purpose, a GNC system consisting of Dubins paths for planning, a side slip compensating line of sight guidance law, a LQR controlling different thrusters, and an UKF for state and current estimation is developed. In total 13 GNC parameters spanning all modules are optimized. This paper uses tuning in simulation to evaluate the versatility of the GNC system and find initial parameters for experiments with different accuracy requirements. Therefore preventing constraint violations during optimization by using safe BO (e.g. [15]) is not required.

Three main results are presented: Each component of the GNC system has a significant influence on overall system performance. Different accuracy requirements and power consumption function can be facilitated with the same policy structure. Robustness can be increased by tuning on a variety of operational scenarios.

This paper is structured as follows: First, in Sec. II the parameter tuning method is introduced. Afterwards, the nanoAUV's model (cf. Sec. III) and its GNC system (cf. Sec. IV) are presented. Sec. V outlines the considered simulation scenario, and results of the optimization are given in Sec. VI.

[1] Institute of Automatic Control, RWTH Aachen University, Aachen, Germany {d.stenger@irt.rwth-aachen.de}

II. PARAMETER TUNING METHOD

A. Optimization problem formulation

Fig. 1 gives an overview over the black-box optimization setting we consider in this contribution. Let $\mathbf{a}$ denote the parameters of the various modules of the GNC system. At each iteration k of the optimization, the closed loop of GNC system with state estimation (filter), planning, guidance, and control as well as the AUV in its environment is simulated with specific GNC parameters $\mathbf{a}_k$ (cf. Sec. IV). Within the simulation the AUV fulfills a set of specific task, in our case following a reference path. The results of the simulation depend on the parameter combination $\mathbf{a}_k$ which is queried by the optimizer. The value of the cost function j_k, in our case total energy consumption, and constraint value g_k, in our case the maximum deviation from the reference path, are returned to the optimizer. Additionally, a binary value l_k is returned which indicates whether the simulation was successful ($l_k = 1$) or not ($l_k = 0$). Reasons for a simulation not being successful are for example diverging state estimation solutions resulting in the AUV not reaching its desired goal state. In this case, returned j_k and g_k values are not meaningful.
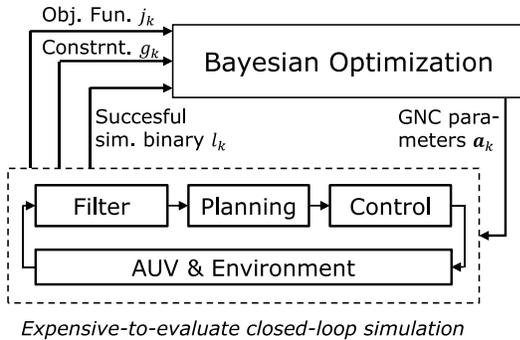

Fig. 1.   Considered black-box optimization setting.

With these definitions we aim at solving the following deterministic constraint optimization problem:

$$\mathbf{a}^\star = \arg\min \quad J(\mathbf{a}) \qquad (1)$$
$$\text{s.t.} \quad \mathbf{a}_{\min} \leq \mathbf{a} \leq \mathbf{a}_{\max}$$
$$g(\mathbf{a}) \leq g_{\max}$$
$$l(\mathbf{a}) = 1$$

Note that the optimization problem is deterministic, because the same draw of disturbance, and model plant mismatch (cf. Sec. V) is used during optimization. In Section VI-D robust optimization is performed on the identical five realizations and therefore still deterministic.

B. Constrained Bayesian optimization (BO) with max-value entropy search and crash constraints

The problem stated in (1) is a real-valued and binary constrained deterministic black-box optimization problem with an expensive-to-evaluate objective function. BO has become a standard tool to solve such problems in the context of control. It can be attributed to the *Micro Data RL* branch of Reinforcement Learning, see [16]. Algorithm 1 gives an overview over BO. For a comprehensive introduction to BO the reader is referred to e.g. [17].

Algorithm 1 Bayesian optimization with crash constraints

1: Initial sampling of A_1, J_1, G_1 and L_1:
2: **for** k = 1; 2; . . . ; **do**
3:     $\tilde{J}_k, \tilde{G}_k \leftarrow$ addArtificialData(A_k, J_k, G_k, L_k)
4:     update probabilistic surrogate models using
      A_k, $\tilde{J}_k$, and $\tilde{G}_k$
5:     select $\mathbf{a}_{k+1}$ by optimizing an acquisition function:
      $\mathbf{a}_{k+1} = \arg\max_{\mathbf{a}}(\alpha(\mathbf{a}|A_k, \tilde{J}_k, \tilde{G}_k))$
6:     query objective function to obtain j_{k+1}, g_{k+1} and l_{k+1}
7:     augment $A_{k+1} = \{A_k, \mathbf{a}_{k+1}\}$, $J_{k+1} = \{J_k, j_{k+1}\}$
      $G_{k+1} = \{G_k, y_{k+1}\}$, $L_{k+1} = \{L_k, l_{k+1}\}$
8: **end for**

At iteration $k = 1$, an initial set of random parametrizations A_1 is generated to obtain initial responses for objective function J_1, constraint G_1 and crash constraint L_1 (cf. Step 1). We use Gaussian process regression (GPR) [18] as the surrogate model. The **surrogate model** uses past evaluations in order to give a probabilistic estimate of the objective function value and the constraint function at unknown locations (cf. Step 4). A separate GP model is created for each response with inputs $\{A_k, \tilde{J}_k\}$ and $\{A_k, \tilde{G}_k\}$, respectively. We use a squared exponential kernel with automated relevance detection and a constant mean function for both models. Since the objective function evaluations are deterministic the noise hyperparameter of the GP model is set to zero. Additionally, a smooth box hyperprior is placed on the kernel length scales in order to avoid very large or very small length scales. The GPs hyperparameters are tuned in each iteration by maximizing the a posteriori likelihood via a combination of random search and gradient decent.

Failed objective function evaluations pose a challenge for BO, because in that case the obtained j and g are not meaningful. This setting is known as **learning under crash constraints** [4]. Setting j and g to some arbitrary large value is not a valid option because this may result in a discontinuous objective function landscapes which contradicts the smoothness assumptions encoded in the Gaussian kernel (cf. e.g. [10]). Instead, here we generate artificial $\tilde{j}$ and $\tilde{g}$ values for the failed simulations (cf. Step 3),

$$\tilde{j} = \mu_j + 3\sigma_j \qquad \tilde{g} = \mu_g, \qquad (2)$$

where μ_j, μ_g, σ_j are the mean and standard deviation of the GP predictions generated ignoring the failed simulations. This approach is similar to the idea in [4], in that it modifies the GP-posterior in a way such it decreases the probability to observe the optimum around the failed observation while not contradicting the smoothness assumptions of the GP.

Constrained max-value entropy search (cMES) [19] is used as the **acquisition function** α because it was shown to outperform the more classical expected improvement with constraints (EIC). cMES chooses the next sample point $\mathbf{a}_{k+1}$, such that the mutual information between the evaluation and the estimated distribution of the optimum is maximized, and is therefore considered to be an information theoretic acquisition function (cf. step 6).

III. THE TRIPLE NANOAUV

The nanoAUV [13], [14] is developed to explore the liquid water column of subglacial lakes. An ice-melting probe transports the nanoAUV to the ice-water interface [20]. The nanoAUV acts as payload, therefore installation space is very limited which restricts the GNC system. As a result, only low performance on-board computers are applicable and the GNC system needs to be developed as simple and computationally efficient as possible.

A. Hydrodynamic model

We follow Fossen et al. [21] in modeling the AUV's 6 degrees of freedom:

$$\mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} + \mathbf{M}_A\dot{\boldsymbol{\nu}}_r + \mathbf{C}_A(\boldsymbol{\nu}_r)\nu_r + \ldots \quad (3)$$
$$\mathbf{D}(\boldsymbol{\nu}_r)\nu_r + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \ ,$$

with gravity $\mathbf{g}(\boldsymbol{\eta})$, mass matrices $\mathbf{M}_{RB}$, and $\mathbf{M}_A$ and coriolis-centripetal matrices $\mathbf{C}_{RB}$, and $\mathbf{C}_A$ for the rigid body (RB) and added mass (A). The state vector $\mathbf{x} = [\boldsymbol{\nu}, \boldsymbol{\eta}]$ consists of the motion components $\boldsymbol{\nu} = [u, v, w, p, q, r]$ in body frame and $\boldsymbol{\eta} = [n, e, d, \phi, \theta, \psi]$ denotes the positions n, e, d of the body frame in the earth fixed frame. The Euler angles are roll ϕ, pitch θ and yaw ψ. The relative velocity $\boldsymbol{\nu}_r = [u - u_c^b, v - v_c^b, w - w_c^b, p, q, r]^T$ is calculated by considering the current vector $[u_c^b, v_c^b, w_c^b]^T$. We use a linear hydrodynamic model without cross couplings and identical drag in sway/heave as well as pitch/yaw movement: $\mathbf{D} = \mathrm{diag}[D_1, D_2, D_2, D_4, D_5, D_5]$.

Because the simulation is considered as a black-box within the optimization, more advanced experimentally validated models can also be used without altering the optimization methodology.

B. Sensors & actuators

The navigation system mainly relies on measurements provided by an IMU, a multi-magnetometer for attitude estimation and an USBL system for positioning. The depth estimation is aided by a pressure sensor. The IMU and magnetometer data are fused within an attitude heading reference system (AHRS) algorithm. The AHRS is not optimized and therefore not discussed in detail in this paper.

Five thrusters are used as the main actuators. The surge thruster with control input u_{surge} is located at the rear of the vehicle, generating a forward pointing force. An additional 4 thrusters are located at the back of the vehicle providing force in the sway (u_{rl}) and heave (u_{ud}) directions. All thrusters except for the surge thruster introduce moments around the center of the vehicle. The thrusters are modeled using a

first order lag element. The generalised forces $\boldsymbol{\tau}$ of Eq. (3) are calculated by combining the geometric properties of the thrusters with the respective control signals. Note that this actuator setup results in an underactuated AUV. Additionally, a buoyancy engine as well as a movable mass for pitch motion are used, influencing $\mathbf{M}_{RB}$ and $\mathbf{C}_{RB}$. They are controlled in a feed-forward manner. Since this feed-forward controller is not optimized, it is not described any further.

IV. GNC SYSTEM

A. State estimation

The state estimator is realized as classical UKF, proposed by [22]. The nonlinear state space model is given by:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{G}\boldsymbol{w} \qquad \boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q}) \qquad (4)$$
$$\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{v} \qquad \boldsymbol{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{R}) \qquad (5)$$

As prediction model $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ the Fossen model from Eq. (3) is used. The actuator control commands act as input $\boldsymbol{u}$. Furthermore, the surge and sway velocities of the current u_c^b and v_c^b are estimated as random walk. The heave component is neglected. This results in the following process model:

$$\begin{pmatrix} \dot{\boldsymbol{\nu}} \\ \dot{\boldsymbol{\eta}} \\ \dot{u}_c^b \\ \dot{v}_c^b \end{pmatrix} = \begin{pmatrix} \text{Fossen from Eq.} (3) \\ \text{Fossen from Eq.} (3) \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} \boldsymbol{w}_\nu \\ \boldsymbol{w}_\eta \\ w_{u_c} \\ w_{v_c} \end{pmatrix}, \qquad (6)$$

with $\boldsymbol{w}_\nu$ and $\boldsymbol{w}_\eta$ as the process noise of the Fossen model and w_{u_c} and w_{v_c} as the process noise of the current model.

For correction, three measurement models $\boldsymbol{h}(\boldsymbol{x})$ are used. The first corrects position $[n, e, d]^T$ with the USBL at a rate of 1 Hz. The second correct makes use of the pressure measurement of the depth sensor at a rate of 10 Hz:

$$z_{press} = k_p \cdot d + p_0 + v_{press}, \qquad (7)$$

with pressure per meter water column k_p, the atmospheric pressure p_0 and the pressure measurement noise v_{press}. The third corrects the attitude $[\phi, \theta, \psi]^T$ using the solution of the AHRS with 100 Hz. We parameterize the process noise covariance matrix $\boldsymbol{Q} = \mathrm{diag}[\boldsymbol{Q}_\nu, \boldsymbol{Q}_\eta, Q_{u_c}, Q_{v_c}]$ with four optimization parameters $\alpha_1, \ldots, \alpha_4$ such that $Q_{u_c} = Q_{v_c} = \alpha_1$, $\boldsymbol{Q}_\nu = [\alpha_2\mathbf{I}^{1\times3}, \alpha_2\alpha_3\mathbf{I}^{1\times3}]^T$ and $\boldsymbol{Q}_\eta = \alpha_4\boldsymbol{Q}_\nu$. The measurement covariance matrix $\mathbf{R}$ is assumed to be known.

B. Path planning

Dubins curves are used to find the shortest path using the current estimated vehicle state and the next waypoint in the horizontal plane. The depth reference is generated by interpolating linearly between starting and goal depth (cf. e.g. [23]). Dynamic vehicle constraints are considered by imposing a minimum turning radius. Waypoints are chosen such that a maximum glide path angle is not superseded. Additionally, a constant surge reference velocity is used. The Dubins curve-based path planner will be used in future work as the local planner within a graph based planning framework. In total, the path planner has two tunable parameters, the reference surge velocity u_{ref} and the Dubins path's radius r_{plan}.

C. Guidance

As mentioned above, the nanoAUV is underactuated. For example, the sway and yaw components cannot be actuated independently from another. In order to resolve the underactuatedness, a line of sight guidance method with adaptive sideslip compensation [24] is used. The main idea is to calculate a reference for the yaw angle, which reduces the horizontal deviation from the reference path using the kinematic relation

$$\psi_d = \gamma_p + \operatorname{atan}(-\frac{h_e}{\Delta}) - \beta_{est}. \qquad (8)$$

The first term in Eq. (8) γ_p represents the path angle and the second term aims at reducing the horizontal crosstrack error h_e (i.e. the shortest distance between the current AUV position and the reference path on the horizontal plane).

The side slip angle $\beta_{est} = \operatorname{atan}(\frac{v}{u})$ occurs due to current induced drift. Here, β_{est} is calculated using the estimated velocities (cf. Sec. IV-A). This way, the state estimation also serves as a disturbance estimator. This approach is similar to estimating β_{est} directly within the guidance module [24]. An analogue approach is used to calculate the pitch reference θ_d.

The quality of the estimation of β is influenced by the parametrization of the state estimation. The main performance influencing tunable parameter within the guidance module is Δ, which determines the aggressiveness of the guidance module.

D. Control

This section formulates a control law for the thrusters. We use the well known LQR method to synthesize a state feedback controller in order to track the surge velocity provided by the planner as well as pitch and yaw angles provided by the guidance. The controlled states are augmented with integral error states u_i, θ_i and ψ_i resulting in an augmented state vector $\mathbf{x}_{fb} = [u, q, r, \theta, \psi, u_i, \theta_i, \psi_i]$. The inclusion of integral error states is necessary in order to achieve offset free tracking in the presence of model plant mismatch. They are calculated by integration of the control error.

The model for LQR synthesis is obtained by discretizing and linearizing Eq. (3). Here we choose a constant working point at $v = 0$ and $u = 0.5\,\mathrm{m/s}$ with all other states set to zero. Potentially beneficial successive linearisation at different working points is not used to save computational resources.

For the time discrete LQR, a state Feedback gain $\mathbf{K}$ is chosen such that the cost function

$$J = \sum_{k=1}^{\infty} \mathbf{x}_{fb}(k)\mathbf{Q}_{LQR}\mathbf{x}_{fb}(k)^T + \mathbf{u}_{lqr}(k)\mathbf{R}_{LQR}\mathbf{u}_{lqr}(k)^T \quad (9)$$

is minimized. The weighting matrices $\mathbf{Q}_{LQR} = \operatorname{diag}[q_1, q_2, q_2, q_3, q_3, q_4, q_5, q_5]$ and $\mathbf{R}_{LQR} = \mathbf{I}$ are diagonal matrices with unknown entries. The LQR control law follows as $\mathbf{u}_{lqr} = [u_{surge}, u_{rl}, u_{ud}] = -\mathbf{K} \cdot \mathbf{x}_{fb}$. Additionally, a combination of a deadband and a hysteresis in order to avoid small high frequency thruster actuation is introduced. In total, 6 tunable parameters $q_1, \ldots, q_5$ and the deadband with w_{db} are considered.

V. SIMULATION SCENARIO

The plant model also relies on Eq. 3. However, in real world operation, model plant mismatch and disturbances deteriorate control and state estimation performance. Therefore, randomly drawn model plant mismatch is introduced in the simulation. The model used within the navigation filter and for controller synthesis deviates from the simulation in the hydrodynamic parameters (e.g. with a standard deviation of 10% for the individual entries of $\mathbf{D}$), and the AUV mass. Additionally, the actuator lag is not considered in filter and controller. Furthermore, the static gain of the thruster transfer function is detuned with a standard deviation of 2.5%. We introduce a randomly generated varying current as an additional disturbance (cf. Fig. 2). For the sensor simulation, USBL, pressure sensor and AHRS signals are perturbed with white Gaussian noise. Additionally, the USBL measurements are delayed by twice the sonic delay. However, the parameter tuning methodology is applicable to other, experimentally validated models, observed mismatches, and observed disturbances.
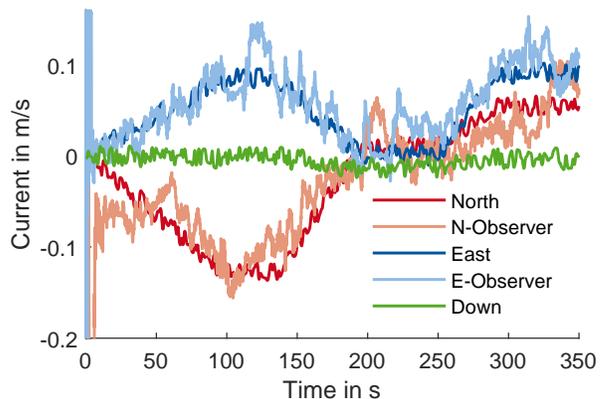

Fig. 2. Randomly generated current profile: Ground truth in dark colors and the observers estimate of the north and east components in light color optimized for maximum controller tracking accuracy.

The reference path is obtained by applying the path planner to a random sequence of way points. With the exception of Sec. VI-D the same seed $s = 1$ for current, way points and model-plant mismatch is used. In these cases the simulated path has a length of $210\,\mathrm{m}$. Except for Sec. VI-C the following arbitrary energy cost function

$$J(\boldsymbol{a}|s) = T_{end}(\boldsymbol{a}|s) + \sum_{i=\{surge, rl, ud\}} \bar{J}_i(\boldsymbol{a}|s) \qquad (10)$$

with

$$\bar{J}_i(\boldsymbol{a}) = \int_{t=0}^{T_{end}(\boldsymbol{a}|s)} |\operatorname{sgn}(u_i(t|\boldsymbol{a}, s))| \cdot \ldots \qquad (11)$$
$$(0.025 + |u_i(t|\boldsymbol{a}, s) + u_i(t|\boldsymbol{a}, s)^{1.5}|)\,\mathrm{dt}$$

is chosen. The time which the AUV needs to reach the final waypoint for a given parametrization $\boldsymbol{a}$ and seed s is denoted as $T_{end}(\boldsymbol{a}, s)$, therefore the first term in Eq. (10) represents consumers with constant power demand such as sensors. The

energy consumed by each thruster is calculated via Eq. (11). Power consumption is only zero if the control input is exactly zero. The maximum observed deviation from the reference path $g(\boldsymbol{a}, s)$ is constrained by $g_{\max} = 1.5\,\mathrm{m}$. In total, one execution of the simulation takes around 30 seconds[1].

VI. Optimization of the GNC System

In the subsequent section, we apply the tuning method to the presented GNC system. It is examined whether the whole GNC system needs to be tuned at once (cf. Sec. VI-A), different accuracy requirements (cf. Sec. VI-B) and different cost functions (cf. Sec. VI-C) require different parametrizations, and (cf. Sec. VI-D robustness can be increased. Each test case is ran 5 times with $45 \cdot d$ objective function evaluations (simulations) each, where d is the number of tunable parameters. For the overall GNC system with 13 parameter one optimization takes around $6.5\,\mathrm{h}$. The tuning parameters for the $\mathbf{Q}$ matrices of state estimation and controller are optimized in the log domain.

A. Joint vs. individual Optimization

Table I compares the combined optimization with the results if only the parameters of the respective GNC submodules are optimized. The default parametrization is hand tuned and does not make use of the deadband. Results indicate that energy consumption can significantly be decreased if the GNC system is optimized as a whole instead of optimizing only one individual component. Additionally, it can be seen that the optimizer finds a good solution reasonably consistent since the deviations of the best and worst run are limited.

TABLE I

INDIVIDUAL VS. JOINT OPTIMIZATION - BEST (WORST) OF FIVE OPTIMIZATION RUNS.

Optimized system	Plan. & Guid.	Control	Filter	Combi.	Default
No. of Params d	3	6	4	13	-
Energy Consumption J	106.6 (111.9)	118.6 (121.1)	135.3 (136.4)	87.6 (94.2)	137.1
maximum Deviation g	1.46 (1.43)	1.49 (1.06)	1.36 (1.22)	1.36 (1.46)	1.11

B. Energy consumption vs. path deviation

In this section, we use the joint optimization of all GNC parameters to examine the conflict of goals between tracking performance and energy consumption. Three different accuracy scenarios are defined. They differ in the maximum allowed planning radius r_{plan} and path deviation requirements $g_{\max}$. In the maximum accuracy scenario, the maximum observed path deviation is minimized in an unconstrained optimization without considering energy consumption. Table II show the results. The conflict of goals becomes apparent especially when comparing the maximum

and medium accuracy scenarios. The low accuracy scenario only slightly decreases energy consumption although the tracking performance is decreased significantly. With decreasing accuracy requirements, we observe an increase in u_{ref}, r_{plan} and Δ which meets expectations. However, for other parameters such as w_{db} and q_4 the trend is not as clear. Additionally, UKF parameters are tuned towards higher position accuracy if better tracking performance is required. In contrast, smoothness in the current estimation is surprisingly not essential for maximum tracking performance (cf. Fig. 2). The best reached tracking accuracy is deemed satisfactory. For collision avoidance, different sensors (e.g. echosounders) and guidance strategies will be used.

TABLE II

DIFFERENT ACCURACY REQUIREMENTS - BEST OF FIVE RUNS.

Scenario	Max. Acc.	Med. Acc.	Low Acc.
Energy consumption J	121.1	87.6	85.2
Max. deviation g	0.82	1.36	2.17
Max. pos. estimation error	0.67	1.10	1.13
Max. allowed r_{plan}	5	10	15
Deviation lim. $g_{\max}$	min!	1.5	3
Optimized u_{ref}	0.47	0.81	0.89
Optimized r_{plan}	4.98	5.74	7.01
Optimized Δ	4.0	59.0	89.3
Optimized $\log 10\,(q_4)$	-3.08	-3.25	-2.88
Optimized w_{db}	0.06	0.13	0.08

C. Varying cost function

The method may also be used to automatically adapt the GNC system for different actuator designs. In order to mimic different actuator designs, we modify the thruster energy function Eq. (11) to a purely quadratic one. Fig. 3 shows the resulting different actuator control signals. Both policies exhibit pulses when following a straight line and a more continuous control input when cornering. However, the optimal time domain behavior changes significantly if different energy consumption characteristics are used. Additionally, the optimized reference velocities u_{ref} is reduced from 0.81m/s (original) to 0.48m/s (quadratic).
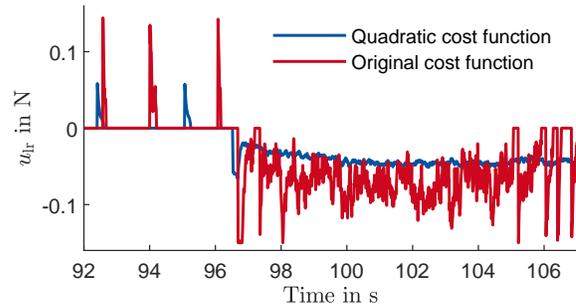

Fig. 3.   Control signal in lateral direction optimized for different power consumption functions. Straight line tracking followed by a right hand curve starting at $96.5\,\mathrm{s}$.

D. Optimization for robustness

Up to now, we only optimized for one fixed model plant mismatch and disturbance trajectory $s = 1$ by using the obj. fun. $J(a|s = 1)$. In practice, the exact model plant mismatch is likely not known a-priori. When validating the Low. Acc. scenario (cf. Sec. VI-B) on 25 different randomly drawn seeds s, eleven violate the path deviation contraint. This indicates that over-fitting is a significant issue, if only one seed is considered for parameter optimization. To tackle this, the same five randomly drawn seeds are evaluated at each objective function query step (cf. Step 6 in Algo. 1). Afterwards, energy consumption is averaged over the five runs $\hat{J}(a) = 1/5 \sum_{s=1}^{5} J(a|s)$ and the maximum path deviation is taken $\hat{g}(a) = \max_{s \in \{1,...,5\}} g(a|s)$. As a result optimization takes five times as long. Apart from that, the optimization problem Eq. (1) is not altered. Results show, that robustness is increased to only one infeasible validation sample at only $\sim 5\%$ more energy consumption on average.

VII. CONCLUSIONS

In this contribution, we have demonstrated the versatile applicability of a BO-based automatic parameter tuning methodology to the holistic optimization of the GNC system of an underactuated AUV. The method can be applied to arbitrary high-level control objectives, vehicles and operational scenarios. As expected, only overall tuning yields the lowest energy consumption. The systematic optimization-based approach, results in a difference of minus 18% compared to the best individual optimization and minus 36% compared to the hand-tuned default parametrization. The method is able to tackle various aspects of the GNC system design procedure such as different actuator designs and robustness. Robustness can be increased drastically by optimizing on various environmental conditions. Furthermore, the method was used to automatically generate different parametrizations for different accuracy requirements. Results indicate, that different requirements can be met with the same GNC structure. This suggests that during operation, different precomputed parameter sets should be used depending on the varying mission requirements.

In future work, the optimization will be extended by an experimentally validated AUV model. Furthermore, it will be investigated in which way the optimized parameters can improve the performance in real experiments.

REFERENCES

[1] X. Xiang, C. Yu, L. Lapierre, J. Zhang, and Q. Zhang, "Survey on fuzzy-logic-based guidance and control of marine surface vehicles and underwater vehicles," *International Journal of Fuzzy Systems*, vol. 20, no. 2, pp. 572–586, 2018.

[2] Z. Zeng, L. Lian, K. Sammut, F. He, Y. Tang, and A. Lammas, "A survey on path planning for persistent autonomy of autonomous underwater vehicles," *Ocean Engineering*, vol. 110, pp. 303–313, 2015.

[3] J. Melo and A. Matos, "Survey on advances on terrain based navigation for autonomous underwater vehicles," *Ocean Engineering*, vol. 139, pp. 250–264, 2017.

[4] A. Marco, D. Baumann, M. Khadiv, P. Hennig, L. Righetti, and S. Trimpe, "Robot learning with crash constraints," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1439–1446, 2021.

[5] A. Marco, P. Hennig, J. Bohg, S. Schaal, and S. Trimpe, "Automatic LQR tuning based on Gaussian process global optimization," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 270–277.

[6] J.-J. Gehrt, W. Liu, D. Stenger, S. Liu and D. Abel, "Environmentally Dependent Adaptive Parameterization of a GNSS-aided Tightly-Coupled Navigation Filter," in *2020 European Navigation Conference (ENC)*. IEEE, 11/23/2020 - 11/24/2020, pp. 1–10.

[7] C. König, M. Khosravi, M. Maier, R. S. Smith, A. Rupenyan, and J. Lygeros, "Safety-aware cascade controller tuning using constrained bayesian optimization," 2020. [Online]. Available: https://arxiv.org/pdf/2010.15211.pdf

[8] T. Herlambang, D. Rahmalia, and T. Yulianto, "Particle Swarm Optimization (PSO) and Ant Colony Optimization (ACO) for optimizing PID parameters on Autonomous Underwater Vehicle (AUV) control system," *J. of Physics: Conference Series*, vol. 1211, p. 012039, 2019.

[9] L. Acerbi and W. J. Ma, "Practical Bayesian Optimization for Model Fitting with Bayesian Adaptive Direct Search," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[10] L. Dörschel, D. Stenger, and D. Abel, "Safe Bayesian Optimisation for Controller Design by Utilising the Parameter Space Approach," in *Proceedings of the 3rd Conference on Learning for Dynamics and Control*, vol. 144. PMLR, 2021, pp. 299–311.

[11] M. Manhães, S. Scherer, L. R. Douat, Martin Voss, and T. Rauschenbach, "Framework for Fair Comparisons of Underwater Vehicle Controllers - Showcasing the Robustness Properties of a Model-free Sliding Mode Controller Tuned with a Random-forest-based Bayesian Optimization Approach," *SIMULTECH 2017*, 2017.

[12] L. Cai, B. Boyacıoğlu, S. E. Webster, L. van Uffelen, and K. Morgansen, "Towards auto-tuning of kalman filters for underwater gliders based on consistency metrics," in *OCEANS 2019 MTS/IEEE SEATTLE*, 2019, pp. 1–6.

[13] C. Waldmann and O. Funke, "The TRIPLE/nanoAUV initiative a technology development initiative to support astrobiological exploration of ocean worlds," *CEAS Space Journal*, vol. 12, no. 1, pp. 115–122, 2020.

[14] M. Nitsch, M. B. Firvida, P. Panten, S. Meckel, T. Koch, D. Stenger, C. Rachuy, J.-J. Gehrt, J. Clemens, and C. Waldmann, "TRIPLE-nanoAUV: A Guidance, Navigation and Control System for an Autonomous Underwater Vehicle to Explore the Oceans of Icy Moons (Poster)," in *18th International Planetary Probe Workshop (IPPW), Session: Innovative Concepts for Exploration*, 2021.

[15] Y. Sui, A. Gotovos, J. Burdick, and A. Krause, "Safe exploration for optimization with gaussian processes," in *Proceedings of the 32nd International Conference on Machine Learning*, 2015, pp. 997–1005.

[16] K. Chatzilygeroudis, V. Vassiliades, F. Stulp, S. Calinon, and J.-B. Mouret, "A survey on policy search algorithms for learning robot controllers in a handful of trials," *IEEE Transactions on Robotics*, vol. 36, no. 2, pp. 328–347, 2020.

[17] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 148–175, 2016.

[18] C. E. Rasmussen and Williams C. K. I., "Gaussian Processes for Machine Learning," 2006.

[19] V. Perrone, I. Shcherbatyi, R. Jenatton, C. Archambeau, and M. Seeger, "Constrained bayesian optimization with max-value entropy search." [Online]. Available: https://arxiv.org/pdf/1910.07003

[20] D. Heinen, P. Linder, S. Zierke, and C. Wiebusch, "An efficient melting probe for glacial research," *Annals of Glaciology*, vol. 62, no. 84, pp. 171–174, 2021.

[21] T. Fossen and A. Ross, "Nonlinear modelling, identification and control of uuvs," in *Advances in Unmanned Marine Vehicles*, G. N. Roberts and R. Sutton, Eds. IET, 2006, pp. 13–42.

[22] E. A. Wan and R. Van Der Merwe, "The unscented kalman filter for nonlinear estimation," in *Proc. of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.

[23] M. Owen, R. W. Beard, and T. W. McLain, "Implementing dubins airplane paths on fixed-wing uavs," in *Handbook of Unmanned Aerial Vehicles*. Springer Netherlands, 2015, pp. 1677–1701.

[24] Fossen Thor I., Pettersen Kristin Y., and Galeazzi Roberto, "Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 820–827, 2015.