2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Controlling an Underactuated AUV as an Inverted Pendulum using Nonlinear Model Predictive Control and Behavior Trees

Sriharsha Bhat[1] and Ivan Stenius[1]

***Abstract*— Agile and hydrobatic maneuvering capabilities can enhance AUV operations in increasingly challenging scenarios. In this paper, we explore the ability of an underactuated AUV to transition to and hold a pitch angle close to 90 degrees at a particular depth, like an inverted pendulum. Holding such an orientation can be valuable in observing a calving glacier, under-ice launch and recovery, underwater docking, inspecting vertical structures, and observing targets above the water surface. However, such control is challenging because of underactuation, rapid response times and varying stability in different configurations. To address this, a control policy is derived offline using nonlinear MPC in a high-fidelity simulation environment in Simulink. For real-time control, a hybrid controller using a behavior tree (BT) is developed based on the optimal MPC policy and applied on the AUV system. The BT controller considers Safety, Transit and Stabilize behaviors. The control algorithm is validated with simulations in Simulink and Stonefish-ROS as well as field experiments with the hydrobatic SAM AUV, showing repeatable performance in the inverted pendulum maneuver.**

I. Introduction

Autonomous underwater vehicles (AUVs) are being used in increasingly challenging scenarios in environmental sensing, aquacultures and security. Slender torpedo-shaped AUVs with simple actuator configurations are usually used in such surveys, with good range but limited maneuverability. Agile and hydrobatic maneuvering capabilities can be beneficial in such applications where such AUVs can then be efficient both in transit as well as in coverage/inspections. Operations in glacier front mapping and under-ice navigation, inspection of remote offshore structures, docking, obstacle avoidance, launch and recovery come with a high degree of uncertainty and risk, and hydrobatic AUV platforms can potentially access previously inaccessible areas [1]–[3]. However, underactuation makes the synthesis of elegant controls challenging, since all states are not directly controllable. Controlling underactuated AUVs has been an open challenge, and several possibilities have been explored in the robotics literature - model predictive control (MPC), reinforcement learning (RL) and hybrid systems have shown recent interest [4]–[7].

In this paper, we explore one such hydrobatic capability- to transition to and hold a pitch angle close 90° vertically at a particular depth. Holding such an orientation can be valuable in aiming sensors, accessing remote areas, and avoiding obstacles in applications including tracking the side of a calving glacier front (Fig. 1), underwater docking (Fig. 2), inspection of vertical structures, under-ice launch and recovery, and even observing targets above the water surface. However, performing such balancing is challenging in real-time, and such operations place high requirements on precision, robustness to disturbances, and safety fallbacks. The system dynamics are nonlinear and heavily dependent on the mass distribution, hydrostatics, hydrodynamic damping, and external forces. Since the system is underactuated, limited states can be controlled. When horizontal, the AUV can be in a statically stable condition with the center of gravity (c.g.) far back and c.g close to the center of buoyancy (c.b), this means the available actuators might not provide enough force to transition from a horizontal to vertical configuration. Once

Fig. 1: *An AUV operating near a calving glacier front in Svalbard, August 2022. Existing control limits make it difficult to approach the glacier.*

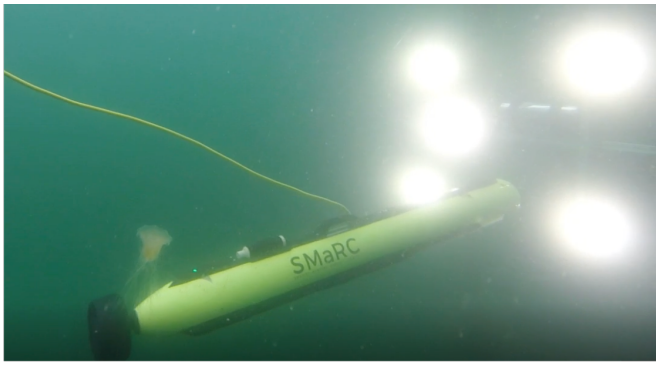


Fig. 2: *The SAM AUV in field tests of underwater docking in Kristineberg, Sweden, June 2022. Vertical docking can be useful in under-ice operations.*

*This work was supported by the Swedish Foundation for Strategic Research (SSF) through the Swedish Maritime Robotics Center (SMaRC).

[1]Sriharsha Bhat and Ivan Stenius are with the School of Engineering Sciences, KTH Royal Institute of Technology, Stockholm, Sweden, svbhat@kth.se and stenius@kth.se

vertical, breaching the water surface and pushing the nose out of the water can suddenly change the c.b. position and the damping, making the system unstable. In the vertical state, the AUV behaves like an *inverted pendulum* and requires local feedback control for stabilization. Furthermore, the transition from the horizontal to vertical state can be quite rapid, adding high requirements on safety and response time.

The inverted pendulum is a classic problem in the control literature, and several approaches including proportional-integral-derivative controllers (PID), linear-quadratic regulators, MPC, RL agents, and hybrid systems have been studied to stabilize inverted pendulums, balance cart-pole systems, and swing up double pendulums [8]–[11]. In some cases, these balancing problems have been combined with robotics, especially quadcopters, mobile robots and motion carriages [12]–[14]. However, an AUV as an underwater inverted pendulum has certain differences from the classical case, with very few published cases in the literature [15]–[18]. The dynamic system is nonlinear and nonholonomic with additional hydrodynamic damping and buoyancy forces. The states have limited reachability due to stability constraints, there are higher environmental disturbances due to waves and currents, and increased sensor uncertainty for state feedback. Furthermore, a transition outside the water surface leads to additional uncertainty due to multi-phase damping and buoyancy forces and limited controllability. Existing approaches to balance AUVs vertically utilized fully actuated or over-actuated systems, thus enlarging the set of reachable states. In [16], a backstepping controller was used to make a hovering AUV pitch up and down. In [15], proportional stabilization algorithms used in quadcopters were translated to a micro-AUV, and the AUV was balanced as an underwater Furuta pendulum. However, when such approaches are translated to underactuated systems, the reachability can be limited, the actuators can saturate before the desired states are reached or certain states may not be controllable. Using model-based techniques such as nonlinear MPC can translate links between actuators and controls and offer elegant control policies for underactuated systems, but at a computational cost that makes them challenging to use in real-time applications.

This paper explores the problem of controlling an underactuated AUV in this inverted pendulum maneuver. The following contributions are made:

1) A control policy is derived offline using nonlinear MPC in a high-fidelity simulation environment.
2) For real-time control, a hybrid controller using a behavior tree (BT) is developed based on the optimal MPC control policy and applied on the AUV system.
3) The BT controller considers *Safety*, *Transit* and *Stabilize* behaviors. Its convergence is discussed.
4) The control algorithm is validated with simulations of a glacier front scenario as well as in field experiments on the Swedish west coast.

Combining offline MPC policies with BTs, relatively simple controllers can be combined to generate complex maneuvering capabilities for underactuated AUVs.

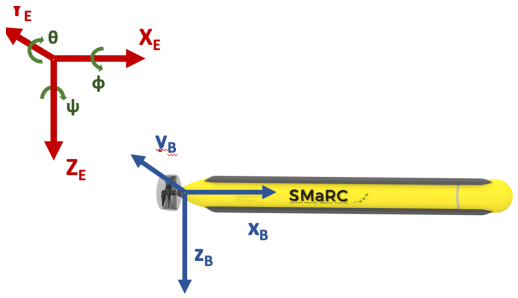


Fig. 3: *AUV world and body reference frames.*

II. System and Dynamics

In this paper, we use the SAM AUV (SAM - Small and Affordable Maritime robot, Fig. 2) developed at KTH within the Swedish Maritime Robotics Center [4], [19] as a case study. SAM has a unique actuator configuration to make it agile, while still being underactuated. Counter-rotating propellers provide thrust forces and moments, while a thrust vectoring nozzle allows the AUV to steer. A movable battery pack enables longitudinal changes to center of gravity (c.g.) position and allows pitch control (LCG), while rotating counterweights allow transversal c.g. changes and offer static roll control (TCG). A variable buoyancy system (VBS) enables static depth control by pumping water in and out of a tank. Navigation and payload sensors include an Inertia Measurement Unit, an Attitude Heading Reference System, a GPS (surface navigation), a Doppler Velocity Logger, cameras, sonar, and environmental sensors (Conductivity-Temperature-Depth probe and chlorophyll fluorometer).

The SAM AUV is modeled with quaternion kinematics and nonlinear dynamics using the notation given by Fossen [20]. A NED coordinate system is used to represent the world frame (with a positive downwards Z_E axis), while a right handed Cartesian coordinate system is used in the AUV frame of reference (Fig. 3). The kinematics in 6DOF are represented by

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J_q}(\boldsymbol{\eta})\boldsymbol{\nu} \ , \tag{1}$$

where $\boldsymbol{J_q} \in \mathbb{R}^{7\times 6}$ is a combined transformation matrix between the pose vector $\boldsymbol{\eta}$ and the velocity vector $\boldsymbol{\nu}$. The pose vector $\boldsymbol{\eta} = \begin{bmatrix} x_E & y_E & z_E & \varepsilon_0 & \varepsilon_1 & \varepsilon_2 & \varepsilon_3 \end{bmatrix}^T$, contains the positions and quaternion orientations in the world frame. Unit quaternions are used instead of Euler angles to avoid a singularity at $90°$ pitch during hydrobatic maneuvers. The velocity vector $\boldsymbol{\nu} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T$ contains the translational and rotational velocities with respect to the x_B, y_B and z_B axes in the body fixed frame.

A vectorial representation is used to describe the dynamics of the AUV as

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}_{\text{eff}}^{-1}(\boldsymbol{\tau_C} - \boldsymbol{C}_{\text{eff}}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{g}(\boldsymbol{\eta})) \ , \tag{2}$$

where $\boldsymbol{M}_{\text{eff}} = \boldsymbol{M_{RB}} + \boldsymbol{M_A}$ and $\boldsymbol{C}_{\text{eff}}(\boldsymbol{\nu}) = \boldsymbol{C_{RB}}(\boldsymbol{\nu}) + \boldsymbol{C_A}(\boldsymbol{\nu}) + \boldsymbol{D}(\boldsymbol{\nu})$ are effective mass and damping matrices. $\boldsymbol{M_{RB}}$ is the rigid body inertia matrix and $\boldsymbol{C_{RB}}$ is the matrix of Coriolis and centripetal

terms. $\boldsymbol{M_A}$ and $\boldsymbol{C_A}(\nu)$ represent the effect of added mass, $\boldsymbol{D}(\nu)$ represents the hydrodynamic damping and $\boldsymbol{g}(\eta)$ is the vector of hydrostatic forces and moments. $\boldsymbol{\tau_C}$ is a vector of external control forces depending on the actuator configuration.

Note that the c.g. position affects the $\boldsymbol{M}_{\text{RB}}$ and $\boldsymbol{C}_{\text{RB}}$ matrices, the c.b. position is affected by the buoyancy forces in $\boldsymbol{g(\eta)}$) and the damping forces in $D(\nu)$ depend on the hydrodynamics. The c.g. position, as well as the external hydrodynamics (or aerodynamics above the surface) can have a major effect on the stability and dynamics of the system.

III. Control strategies

A. Nonlinear MPC to generate a control policy offline

Model Predictive Control (MPC) can be used to generate an optimal control policy for an underactuated AUV, and the nonlinear system's limits can be explored. MPC offers a pragmatic approach to solving an optimal control problem through a receding horizon strategy considering a pre-defined prediction horizon. At each time step, the optimal control is recalculated for the prediction horizon considering an objective function, dynamics model, and constraints, and the result is applied to the system. This receding horizon strategy reduces computational effort (due to the finite horizon) while adding some robustness to disturbances (thanks to the recalculation of the optimal control).

A nonlinear model predictive control (NMPC) problem is defined for output reference tracking (where the optimal control must minimize the deviation of the output from a reference value) using the dynamics model presented in (2) in the previous section. The output state is given by $\boldsymbol{s}_{\text{out}}^{\boldsymbol{T}} = \begin{bmatrix} \boldsymbol{\eta}_{\text{out}} & \boldsymbol{\nu}_{\text{out}} \end{bmatrix}^T$, while the reference state to be tracked is given by $\boldsymbol{s}_{\text{ref}}^{\boldsymbol{T}} = \begin{bmatrix} \boldsymbol{\eta}_{\text{ref}} & \boldsymbol{\nu}_{\text{ref}} \end{bmatrix}^T$. We assume the vector of control forces $\boldsymbol{\tau_C}$ to be a function of the vector of available control inputs $\boldsymbol{c}$.

Problem 1. *The NMPC problem is formulated as*

$$\begin{aligned} \min_{\boldsymbol{\tau_C}} \quad & J = \textstyle\int_t^{t+T_p} [(\boldsymbol{s}_{ref} - \boldsymbol{s}_{out})^T \boldsymbol{Q} (\boldsymbol{s}_{ref} - \boldsymbol{s}_{out}) + \\ & \boldsymbol{c}^{\boldsymbol{T}} \boldsymbol{R_1} \boldsymbol{c} + \dot{\boldsymbol{c}}^{\boldsymbol{T}} \boldsymbol{R_2} \dot{\boldsymbol{c}}] dt \\ s.t. \quad & \\ & (1), \\ & (2), \\ & \boldsymbol{s}_{out} = \boldsymbol{o(\eta, \nu)}, \\ & \boldsymbol{s}_{out} \in \mathbb{S}, \\ & \boldsymbol{\tau_C} \in \mathbb{T}. \end{aligned}$$

where t is the time, T_p is the prediction horizon, and $\boldsymbol{Q}, \boldsymbol{R_1}$ and $\boldsymbol{R_2}$ are weighting matrices. The state dynamics are given by equations (1) and (2), and are subject to state and control constraints contained in sets $\mathbb{S}$ and $\mathbb{T}$.

The MPC aims to minimize the deviation from the reference trajectory, while also minimizing the control input $\boldsymbol{c}$ and its rate $\dot{\boldsymbol{c}}$. In the case of an inverted pendulum maneuver, we want to prescribe a reference orientation close to -90° pitch, and a reference depth, and prioritize these states in $\boldsymbol{Q}$ along with the propellers and thrust vectoring in $\boldsymbol{R_1}$ and $\boldsymbol{R_2}$. Due to the large state space and rapid maneuvering requirements, the NMPC solver requires significant computational cost for real-time control. This motivates computationally simpler control approaches based on the NMPC control policy.

B. Hybrid control using BTs to execute the policy online

Given the nonlinear nature of the system and high real-time requirements, a hybrid control strategy can be beneficial for online implementation. Behavior trees (BT) can offer a modular and robust framework for such hybrid control, enabling goal-based and deliberative task execution [21]. A definition of BTs follows [22],

Definition 1. *A BT is a three-tuple*

$$\mathcal{T}_i = \{f_i, r_i, \Delta t\} \tag{3}$$

where $i \in \mathbb{N}$ is the index of the tree, $f_i : \mathbb{R}^n \to \mathbb{R}^n$ is the right hand side of an ordinary difference equation, Δt is a time step and $r_i : \mathbb{R}^n \to \{R, S, F\}$ is the return status that can be equal to running, success or failure. The three regions $R_i, S_i, F_i \subset \mathbb{R}^n$ of $\mathcal{T}_i$ are defined as

$$\begin{aligned} R_i = & \quad \{x : r_i(x) = R\} \\ S_i = & \quad \{x : r_i(x) = S\} \\ F_i = & \quad \{x : r_i(x) = F\} \end{aligned} \tag{4}$$

The execution of a BT is a standard ordinary difference equation

$$\begin{aligned} x_{k+t}(t_{k+1}) &= f_i(x_k(t_k)) \\ t_{k+1} &= t_k + \Delta t \end{aligned} \tag{5}$$

Based on this definition, we propose a *Safety-Transit-Stabilize Behavior Tree (STS-BT)*, synthesized based on the MPC policy, see Fig. 4:

1) **Safety:** Perform safety actions in case of emergency (return regions R_1, S_1, F_1).
2) **Transit:** Transit to a stabilizable region (return regions R_2, S_2, F_2).
3) **Stabilize:** Run stabilizing controller in the stabilizable region (return regions R_3, S_3, F_3).

Such a framework can be generalized for several hydrobatic maneuvers. During transit, an open-loop motion primitive is used to rapidly guide the system to a region of attraction S_2. Within this controllable region, the control switches to a stabilizing feedback controller (in R_3), balancing the AUV at the reference state and holding the system within the region of attraction S_3. The feedback controller is a PID regulator in this case (but can even be MPC or an RL agent in others). Also, in more complex maneuver sequences, it might be necessary to use more advanced controllers (or sub-trees) for transit. If safety limits are violated, recovery actions are performed as fallbacks (in F_1).

C. Convergence

A key design objective in controller design is to make the system stable or asymptotically stable with respect to an equilibrium point. First, considering the offline policy with

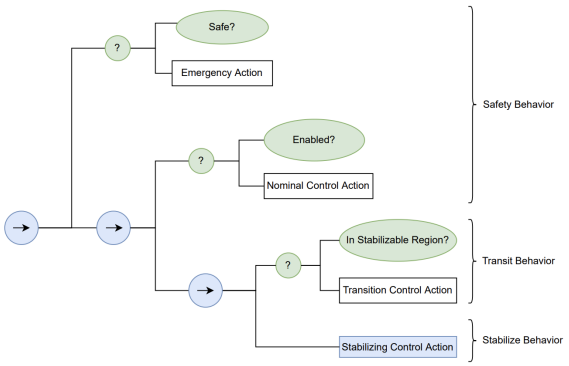


Fig. 4: *The Behavior Tree for hydrobatics. Here the AUV passes safety checks, is in the stabilizable region and is running the Stabilize control action. Nodes include sequences (arrows), fallbacks (question marks), actions (rectangle) and conditions (ellipse). All nodes can return Success (green), Failure (red) and Running (blue).*

NMPC, if the state constraints constrain the state within a region of attraction Ω, then with $\Omega \subseteq \mathbb{S}$, the formulation shown in *Problem 1* can drive the system to an equilibrium. Second, from *Definition 1* above, the STS-BT is a hybrid controller with sets S_i, R_i, F_i that represent *running*, *success* and *failure* regions respectively for each subtree. The design objective is for each sub-tree is to reach the success region S_i in finite time. Otherwise, if the subtrees are in a sequential composition and the first converges to S_1 as $t_1 \to \infty$, the other subtrees would never execute, and the entire sequence of controllers would not succeed.

Definition 2. *A BT is Finite Time Successful (FTS) with region of attraction R', if for all starting points $x(0) \in R \subset R$, there is a time τ such that $x(\tau) \in S$ for some $\tau' \leq \tau$ and $x(t) \in R'$ for all $t \in [0, \tau')$] [22])*

Given the right choice of the sets S_i, F_i, R_i, exponential stability of the individual controllers implies Finite Time Success, according to *Lemma 1*. This implies that, if each individual controller within the BT is designed to be exponentially stable, the entire composition converges to a successful state in finite time.

Lemma 1. *A BT for which x_s is a globally exponentially stable equilibrium of the execution (5), and $S\{x : ||x - x_s|| \leq \epsilon\}, \epsilon > 0, F = \emptyset, R = \mathbb{R}^n \backslash S$ is Finite Time Successful*

Proof: See [22]

IV. Results

A. Reachability

The AUV system's reachable states are plotted in Fig. 5, where random controls are applied to the system for 30 seconds from an initial position on the origin. It can be seen that all states are not reachable from the control configuration, and to transition to 90° pitch angle at a particular depth, pre-defined maneuver sequences can be beneficial.

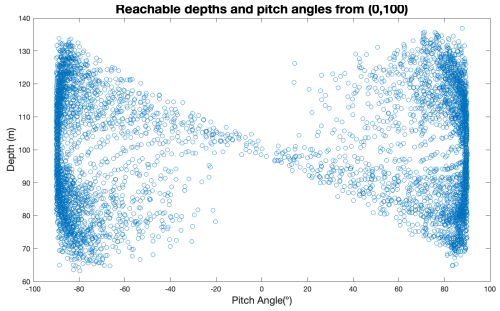


Fig. 5: *Reachable states beginning from 100m depth, 0 degrees pitch, 5000 random controls.*

B. Policy generation and controller synthesis in Simulink

The optimal control policy for the inverted pendulum maneuver is derived using NMPC in a high fidelity simulation environment for hydrobatics in Simulink. SAM is modeled using a multi-fidelity hydrodynamic database which gives relatively accurate dynamic behavior at high angles of attack (for more information on the simulator, see [23]). *Problem 1* is implemented in Matlab and solved using the Sequential Quadratic Programming algorithm in Mathworks' Nonlinear MPC toolbox with the following parameters:

$$\begin{array}{ll} \boldsymbol{s}_{\text{ref}} = & \begin{bmatrix} 0 & 0 & 19 & 0 & 0 & \text{deg2rad}(80°) & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ \boldsymbol{Q} = & \text{diag}(\begin{bmatrix} 100 & 100 & 100 & 100 & 100 & 100 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}) \\ \boldsymbol{R_1} = & \text{diag}(\begin{bmatrix} 100 & 100 & 10 & 10 & 1000 & 1000 \end{bmatrix}) \\ \boldsymbol{R_2} = & \text{diag}(\begin{bmatrix} 100 & 100 & 10 & 10 & 1000 & 1000 \end{bmatrix}). \end{array} \tag{6}$$

The results of using NMPC to control SAM are presented in Fig. 6. The control objective was to start at 20m depth and 0° pitch, and transit to a vertical configuration (close to $-80°$ or -1.35rad pitch) and maintain a depth setpoint of 19m. The propellers and thrust vectoring were prioritized with higher weights in the $\mathbf{R_1}, \mathbf{R_2}$ weight matrices in (6). The simulation was run for 100 seconds. It can be seen that the AUV holds the pitch angle steadily with some oscillation in depth (with an overshoot of 0.3m). A bang-bang sequence is used in thrust vectoring, the propellers are used actively and the trim subsystems are maintained at a constant level (VBS at neutral buoyancy and LCG setting the c.g. back as far as possible). Looking at the control policy, at the start, a sequence with the LCG all the way back, negative propeller rpm, downwards thrust vectoring and neutral VBS is used to transit to a stabilizable configuration, following which propellers and thrust vectoring are used actively to hold the AUV in position. This optimal policy for the inverted pendulum maneuver is then implemented within the STS-BT with the following subtrees:

1) **Safety:** SAM performs emergency surfacing. The VBS is set to minimum buoyancy and all other actuators are disengaged. This action is triggered if SAM dives beyond a safe depth. If the controller is disengaged, nominal controls are sent to all actuators and SAM hovers in place.
2) **Transit:** Based on the MPC policy, an open loop reverse-diving sequence is applied. The action sets VBS and LCG to constant values, sets a negative propeller rpm and uses downwards thrust vectoring

Metric	NMPC	STS-BT
Rise-time (s)	20.7	25.0
Settling time (s)	34.5	50.0
Steady-state error (%)	0.9	2.9
Overshoot (%)	4	6.5

TABLE I: *Performance comparison of the NMPC and STS-BT controllers for 4 setpoints in Simulink, average values.*

to switch to a vertical configuration while diving backward.

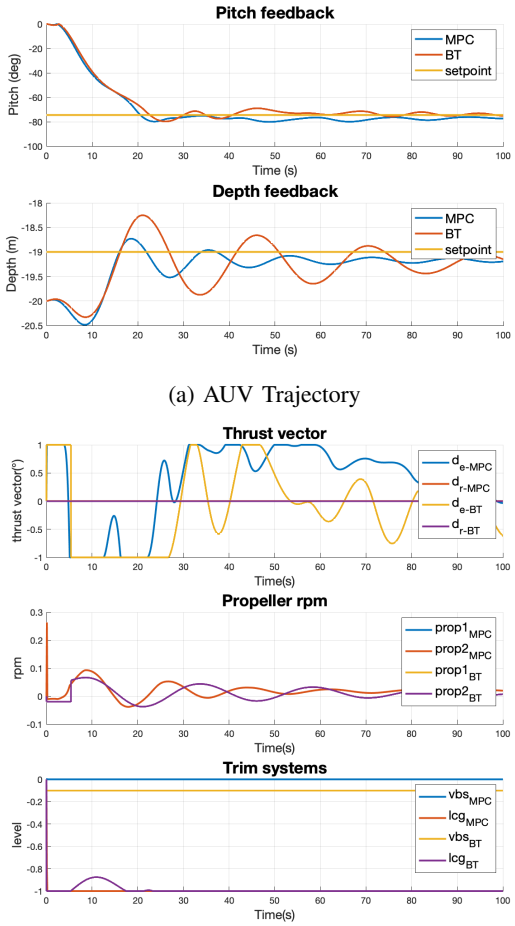
3) **Stabilize:** Once the pitch angle crosses a threshold ($-10°$ in this case), the *Stabilize* controller takes over. This consists of PID controllers that regulate the depth using the propeller rpm, and the pitch and yaw angles using thrust vectoring, while the LCG and VBS remain at constant values.

The results of the STS-BT controller are presented in the same plot in Fig. 6, and it can be seen that the hybrid controller has qualitatively similar (but suboptimal) behavior to the MPC while holding a pitch angle close to $-80°$, with a higher overshoot in depth (0.6m instead of 0.3m) and a similar rise time of 21 seconds. Furthermore, on running both the MPC and the STS-BT controller for a longer sequence of 200s with 4 different depth setpoints, it was seen that the BT controller and the MPC had quantitatively similar performance in terms of overshoot, rise-time, and steady state error (see Table I) though the STS-BT had a higher settling time and overshoot. The AUV in the simulation environment can be seen in Fig. 7 a. Also, note that while the BT could be run in real-time, the nonlinear MPC simulation required 900 seconds to simulate a 200s maneuver, with a real-time ratio of 4.5.

C. Field Experiments with the SAM AUV

The autonomy software on SAM runs in the Robot Operating System (ROS) environment, and the STS-BT framework was implemented in ROS. While the simulator in Simulink enables detailed testing of control strategies with high-fidelity dynamics, the Stonefish simulator in ROS allowed validation of higher level software and rehearsal of mission scenarios in a virtual world with photorealistic features, albeit with lower fidelity dynamics. Before deployment on the real AUV, the BT software was first validated in a glacier front scenario within the Stonefish simulator [25] (see Fig. 7b). In this scenario, the AUV started on the surface, dived backwards to 2m depth to reach a vertical configuration, and then came back up to the surface to peek out of the water.

Following validation in Stonefish, the STS-BT was tested and optimized on the SAM AUV hardware through field experiments at research stations in Djurö in the Stockholm archipelago and Kristineberg on the Swedish west coast (Fig. 7c). To account for differences between the simulation model and the real AUV, the threshold pitch angle and the reverse rpm were increased in the *Transit* subtree, and the PID coefficients were tuned to reflect real-world dynamics in the *Stabilize* subtree. The results from a 7.5-minute sequence



(a) AUV Trajectory

(b) Normalized Actuator Commands

Fig. 6: *Results of balancing in Simulink using nonlinear MPC and the STS-BT controller, 100s.*

holding a pitch angle close to $-80°$ and switching between different depth setpoints are presented in Fig. 8. The AUV tracked the pitch setpoint while holding the depth within a steady state error and overshoot of 0.4m. Due to the high propeller rpm, the maneuver studied was quite fast, dynamic and sensitive to disturbances — this meant the *Safety* behavior with a depth limit was beneficial. The AUV breached the water surface and peeked outside between 150-200s (this can be seen with the reduced and oscillating pitch angle). Between 350-400s, the PID gains were increased to use rpms, leading to higher pitch angles close to $-90°$(but with higher oscillations).

V. Discussion

In the simulations, NMPC took advantage of the dynamics of the underactuated system and computed elegant control policies that balanced setpoint tracking with actuator usage. However, for complex and high-speed real-time maneuvers such as this, it was not computationally feasible to solve the NMPC optimization problem online — a hybrid control approach was preferred. The STS-BT shows qualitatively and quantitatively similar performance to the NMPC policy. While it requires some nuance to explore the reachable states

(a) Simulink (b) Glacier front in Stonefish (c) Field tests at Kristineberg

Fig. 7: *SAM performing the 'Inverted Pendulum' maneuver in simulations and field tests. Videos are available in [24].*

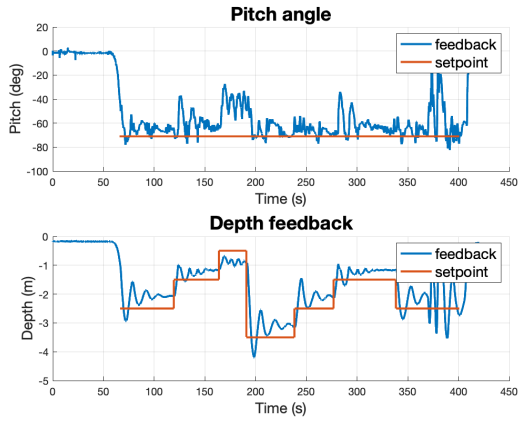


Fig. 8: *Results from field tests using the STS-BT controller.*

and find the stabilizable region S_2, the MPC policy offers good guidance for defining the *Safety*, *Transit* and *Stabilize* subtrees. Translating the NMPC policy to a hybrid system (STS-BT) is a key contribution of this paper. Considering convergence — the *Stabilize* PID controller is theoretically stabilizing, and the *Safety* and *Transit* controllers will drive the system to either a safe or stable region.

The STS-BT performed well in field tests with the SAM AUV. SAM was able to hold a vertical orientation and switch between different depths. Despite a change in the system dynamics on hitting the water surface, SAM was also able to stabilize itself, albeit with oscillations. The *Stabilize* PID controller was relatively robust against disturbances and was not difficult to tune. The *Transit* phase while diving backward at high speed was challenging and the presence of the *Safety* subtree was useful. However, the actual AUV had higher actuator saturation limits and a more stable c.g. position than the simulation model. SAM required very high rpms to cross the 90° pitch threshold, and it was challenging to keep the AUV stationary in that configuration. This meant that a balance had to be struck between smooth control and aggressive maneuvering. Current work focuses on improving controller performance (in terms of oscillations and steady state error), using more advanced controllers for stabilization under high oscillations (e.g with real-time MPC), and making the simulation more realistic (e.g. through system identification). This will be crucial for detailed simulations and experiments including payload sensors in scenarios such as docking, inspection, and glacier mapping.

A key feature of this method is to combine multiple simple controllers in a safe manner. Automating subtree and controller synthesis based on the optimal control policy can be valuable moving forward. Further, this control approach could be applied to other hydrobatic maneuvers by encapsulating control strategies such as real-time MPC, LQR and RL in the STS-BT framework. The *Safety* subtree allows one to use more aggressive control in other subtrees without fear of damage. However, it is important to understand the system dynamics to generate the subtrees — some maneuvers (such as performing vertical looping or tight turning) might have a complex transit region and a very small stabilizable region or vice versa. Additionally, theoretical analysis on BT convergence such as [26], [27] can also be beneficial for generalization.

VI. Conclusion

This paper combined nonlinear model predictive control and behavior trees to control an underactuated AUV. The inverted pendulum maneuver was studied, where the slender and hydrobatic SAM AUV had to transition to and hold a vertical pitch angle close to -90° at a specific depth. NMPC was used to derive a control policy offline in simulation. A real-time BT controller (called the STS-BT) was synthesized from the NMPC policy, considering *Safety*, *Transit* and *Stabilize* subtrees. The STS-BT controller showed similar behavior to NMPC in high-fidelity simulation in terms of actuator usage, rise time, overshoot and steady-state error. The controller was demonstrated in field tests with the SAM AUV, where the STS-BT had good and repeatable performance. Next steps will focus on control in conjunction with payload sensors for use-case scenarios such as docking, glacier mapping, and structure inspections.

Acknowledgement

We thank Andreas Ledzins, Josefine Severholt, Carl Ljung and Koray Amico Kulbay for their support in the field tests.

REFERENCES

[1] S. Bhat and I. Stenius, "Hydrobatics: A review of trends, challenges and opportunities for efficient and agile underactuated AUVs," in *2018 IEEE/OES AUV 2018 , Porto*, Nov 2018.

[2] I. Stenius, J. Folkesson, S. Bhat, C. I. Sprague, L. Ling, Ö. Özkahraman, N. Bore, Z. Cong, J. Severholt, C. Ljung, A. Arnwald, I. Torroba, F. Gröndahl, and J.-B. Thomas, "A system for autonomous seaweed farm inspection with an underwater robot," *Sensors*, vol. 22, no. 13, 2022.

[3] P. King, G. Williams, R. Coleman, K. Zürcher, I. Bowden-Floyd, A. S. Ronan, C. Kaminski, J.-M. Laframboise, S. D. McPhail, J. P. Wilkinson, A. D. Bowen, P. Dutrieux, N. Bose, A. K. Wåhlin, J. Andersson, P. Boxall, M. Sherlock, and T. Maki, "Deploying an auv beneath the sørsdal ice shelf: Recommendations from an expert-panel workshop," *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, pp. 1–6, 2018.

[4] S. Bhat, I. Torroba, Özkahraman, N. Bore, C. I. Sprague, Y. Xie, I. Stenius, J. Severholt, C. Ljung, J. Folkesson, and P. Ögren, "A cyber-physical system for hydrobatic AUVs: System integration and field demonstration," in *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, 2020, pp. 1–8.

[5] R. M. Saback, A. G. S. Conceicao, T. L. M. Santos, J. Albiez, and M. Reis, "Nonlinear model predictive control applied to an autonomous underwater vehicle," *IEEE Journal of Oceanic Engineering*, vol. 45, no. 3, pp. 799–812, 2020.

[6] L. V. Steenson, S. R. Turnock, A. B. Phillips, C. Harris, M. E. Furlong, E. Rogers, L. Wang, K. Bodles, and D. W. Evans, "Model predictive control of a hybrid autonomous underwater vehicle with experimental verification," *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, vol. 228, no. 2, pp. 166–179, 2014.

[7] S. Heshmati-alamdari, G. C. Karras, P. Marantos, and K. J. Kyriakopoulos, "A robust model predictive control approach for autonomous underwater vehicles operating in a constrained workspace," 2018.

[8] Y. Xin, J. Xu, B. Xu, and H. Xin, "The inverted-pendulum model with consideration of pendulum resistance and its lqr controller," in *Proceedings of 2011 International Conference on Electronic Mechanical Engineering and Information Technology*, vol. 7, 2011, pp. 3438–3441.

[9] T. Ohhira and A. Shimada, "Model predictive control for an inverted-pendulum robot with time-varying constraints," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 776–781, 2017, 20th IFAC World Congress.

[10] J. Shen, A. Sanyal, N. Chaturvedi, D. Bernstein, and H. McClamroch, "Dynamics and control of a 3d pendulum," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 1, 2004, pp. 323–328 Vol.1.

[11] H. Igarashi, T. Saito, T. Kinjyo, and F. Matsuno, "Development of an autonomous inverted pendulum mobile robot for outdoor environment," in *2008 SICE Annual Conference*, 2008, pp. 2282–2285.

[12] M. Hehn and R. D'Andrea, "A flying inverted pendulum," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 763–770.

[13] R. Figueroa, A. Faust, P. Cruz, L. Tapia, and R. Fierro, "Reinforcement learning for balancing a flying inverted pendulum," in *Proceeding of the 11th World Congress on Intelligent Control and Automation*, 2014, pp. 1787–1793.

[14] H. Lee and J. Lee, "Driving control of mobile inverted pendulum," in *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2012, pp. 449–453.

[15] D. A. Duecker, A. Hackbarth, T. Johannink, E. Kreuzer, and E. Solowjow, "Micro underwater vehicle hydrobatics: A submerged furuta pendulum," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7498–7503.

[16] B. M. Ferreira, J. Jouffroy, A. C. Matos, and N. A. Cruz, "Control and guidance of a hovering auv pitching up or down," in *2012 Oceans*, 2012, pp. 1–7.

[17] A. Astolfi, G. Picardi, and M. Calisti, "Multilegged underwater running with articulated legs," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1841–1855, 2022.

[18] S. Hasnain, U. H. Shah, S.-H. Choi, and K.-S. Hong, "Dynamics and vibrational control of an underwater inverted pendulum," in *2016 16th International Conference on Control, Automation and Systems (ICCAS)*, 2016, pp. 644–649.

[19] S. Bhat, I. Stenius, N. Bore, J. Severholt, C. Ljung, and I. Torroba Balmori, "Towards a cyber-physical system for hydrobatic auvs," in *OCEANS 2019 - Marseille*, June 2019, pp. 1–7.

[20] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons Ltd., April 2011.

[21] A. Marzinotto, M. Colledanchise, C. Smith, and P. Ögren, "Towards a unified behavior trees framework for robot control," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 5420–5427.

[22] M. Colledanchise and P. Ögren, "How behavior trees modularize robustness and safety in hybrid systems," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1482–1488.

[23] S. Bhat, I. Stenius, and T. Miao, "Real-time flight simulation of hydrobatic AUVs over the full 0–360 envelope," *IEEE Journal of Oceanic Engineering*, vol. 46, no. 4, pp. 1114–1131, 2021.

[24] SMaRC. Hydrobatics with the SAM AUV as an inverted pendulum. [Online]. Available: https://www.youtube.com/watch?v=AQMXcfKdioc

[25] P. Cieślak, "Stonefish: An Advanced Open-Source Simulation Tool Designed for Marine Robotics, With a ROS Interface," in *OCEANS 2019 - Marseille*, Jun. 2019.

[26] P. Ögren, "Convergence analysis of hybrid control systems in the form of backward chained behavior trees," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6073–6080, 2020.

[27] C. I. Sprague and P. Ögren, "Continuous-time behavior trees as discontinuous dynamical systems," *IEEE Control Systems Letters*, vol. 6, pp. 1891–1896, 2022.