Robot Vitals and Robot Health: Towards Systematically Quantifying Runtime Performance Degradation in Robots Under Adverse Conditions

Aniketh Ramesh, Rustam Stolkin, and Manolis Chiou

***Abstract*—This letter addresses the problem of automatically detecting and quantifying performance degradation in remote mobile robots, in real-time, during task execution. A robot may encounter a variety of uncertainties and adversities during task execution, which can impair its ability to carry out tasks effectively and cause its performance to degrade. Such situations can be mitigated or averted by timely detection and intervention, e.g., by a remote human supervisor taking over control in teleoperation mode. Inspired by patient triaging systems in hospitals, we introduce the framework of "robot vitals" for estimating overall "robot health". A robot's vitals are a set of lower-level metrics that estimate a variety of indicators of performance degradation faced by a robot at any given point in time. Robot health is a higher-level metric that combines robot vitals into a single scalar value estimate of performance degradation. Experiments, both in simulation and on a real mobile robot, demonstrate that the proposed robot vitals and robot health can be used effectively for online estimation of robot performance degradation during run-time.**

***Index Terms*—Failure detection and recovery, human-robot teaming, robotics in hazardous fields.**

I. Introduction

ROBOTS operating in remote environments, regardless of their advanced capabilities, often face various issues and performance degradation during task execution. This is particularly true for tasks in extreme environments, where the robots are typically physically remote from a human operator who remains in a safe zone. Examples of performance degrading factors include terrain adversities, camera occlusion, sensor noise, limitations in AI capabilities, and unexpected circumstances. Irrespective of the control mode used (e.g., full autonomous or teleoperated Robots, Shared Control, Variable Autonomy), when robots are subjected to such factors for a prolonged period, they may behave unpredictably, perform tasks sub-optimally, or fail catastrophically.

Manuscript received 22 February 2022; accepted 27 June 2022. Date of publication 20 July 2022; date of current version 9 August 2022. This letter was recommended for publication by Associate Editor F. Pierri and Editor C. Gosselin upon evaluation of the reviewers' comments. This work was supported by NCNR under Grants EP/R02572X/1, EP/P01366X/1, and EP/P017487/1, and in part by ReLiB under Grant FIRG005.

The authors are with the Extreme Robotics Lab (ERL), University of Birmingham, B29 7DE Birmingham, U.K. (e-mail: axr1050@student.bham.ac.uk; r.stolkin@bham.ac.uk; m.chiou@bham.ac.uk).

Digital Object Identifier 10.1109/LRA.2022.3192612

Robot performance degradation is commonly mitigated by changing the control mode, or by initiating pre-programmed recovery behaviours. While it is possible for a remote human supervisor to detect problems and intervene [1], it would also be useful to provide robots with a means to reliably detect performance degradation automatically. This would enable autonomous robots to trigger automatic recovery behaviours or call for human intervention [2].

In this study, the term "performance degradation" refers to any impairment in the capability of a robot to carry out its tasks. Automatically detecting situations where a robot is facing performance degradation in real-time, is a challenging open problem. Any such framework requires metrics applicable to a wide range of robots; to quantify the effects of hardware errors, software limitations, and environmental factors, during problematic situations. The framework should also be robust, and easily adaptable to constraints imposed by the robot's sensors, morphology or underlying algorithms used.

This letter proposes a framework to detect and quantify robot performance degradation during task execution, simultaneously for different levels of abstraction, by using a set of performance indicators called the "robot vitals". The "robot health" is a meta-metric that combines all vitals into a single scalar value, estimating the intensity of performance degradation. Using such a framework, an AI agent can estimate the performance degradation that a robot is facing, communicate (e.g. via visual cues [3]) and call for operator assistance for semi-autonomous robots [4], trigger recovery behaviours automatically, or even suggest actions to mitigate the effect of problematic situations. The design of this framework is inspired by the simplicity and standardisation of patient triaging systems used by hospitals, where threshold values of the classic "vital signs" physiological parameters (e.g., pulse, temperature, blood pressure, and oxygen saturation) are used to assign a clinical health intensity score to patients [5].

Estimating performance degradation can also inform AI policies on shared control and/or mixed-initiative control systems (i.e. systems in which both humans and robots are capable of seizing or relinquishing control of different mission elements [2], [6]). This is particularly important in multi-robot systems, as a single operator may be required to simultaneously monitor several robots with limited cognitive resources, leading to high cognitive workload and sub-optimal assistance [7].

For these reasons, our experiments test the framework on a mobile robot that is navigating fully autonomously throughout each experiment, while encountering challenging situations. However, in the future we intend to use this system to trigger autonomous recovery behaviours or inform Level of Autonomy (LoA) switching systems. We hypothesise that by using robot vitals and robot health, the runtime performance degradation of robots can be estimated.

The main contributions of this letter are: a) proposing and introducing the framework of robot vitals and robot health; b) proposing a realisation with a set of five example robot vitals and an intuitive scalable meta-metric that combines the robot vitals to calculate a robot's health; c) presenting and examining the results of systematic experiments on the robot vitals and robot health framework, applied to an autonomous mobile robot.

II. Related Work

Robot performance degradation that can be mitigated, without removing the robot from the operating environment, is called a field repairable failure [8]. Such failures are non-terminal [8] as they only cause a temporary lapse in task execution. However, if left unattended, non-terminal failures can become terminal. Some factors responsible for such failures are sensor noise, unpredictable robot behaviour, wheel encoder faults, motor malfunctions, communication losses and sudden power drops [9]–[11].

Literature on performance evaluation commonly describes metrics that can be calculated offline before or after the robot has finished operating [12]. These metrics typically quantify: a robot's task performance, e.g. time to task completion; reliability e.g., Mean Time to Failure and Mean Time Between Failures [10], [13]; or qualitative risk (low, medium, or high risk) posed by a robot to its operators and surroundings [14], [15]. In contrast, this letter addresses the problem of real-time detection, quantification and monitoring of robot performance degradation, online during missions, to help detect and overcome problematic situations.

The most common method of online detection of hardware or electronic issues related to robot movement are control systems and dead-reckoning systems [16]. Such systems trigger an error signal if a robot deviates from a pre-defined model of ideal robot behaviour. Alternatively, runtime performance degradation can be detected using heuristics defined on threshold values of sensor data. These heuristics may use differences between: localisation estimates [17]; the robot's actual versus ideal velocity given by an expert planner [2]; expected versus actual time to complete a task [17], [18]; a robot's mean velocity, displacement or the total area explored [19]. Alternatively, online metrics can also be extracted by machine learning of various types [20], [21] but this requires large data sets of robot failures [22] to be available or created in simulation. In summary, the letters cited above use task-specific online metrics to detect poor robot performance. They do not propose a general framework for measuring degradation, and do not explicitly examine the relationship between such metrics and overall robot performance degradation.

Our previous work [23] gave a rudimentary introduction to the concept of robot vitals and robot health. Here we present a more systematic approach for reasoning about the vitals, and an experimental validation of the vitals and overall health metrics. Our aim is to begin addressing a gap in the literature on quantifying robot performance degradation online in real-time during missions. This letter demonstrates these concepts in a particular example robot and mission scenario, but we hope to also provide some fundamental and generalisable insights that can help roboticists choose appropriate vitals metrics for other types of robots and tasks.

III. Robot Vitals

Robot vitals are online, real-time metrics that indicate performance degradation faced by a robot at any time during a its mission. In contrast to task performance metrics, vitals should indicate the degree to which a robot is continuing (or unable) to function during adverse conditions without failing or behaving erroneously. Each vital represents a specific aspect of robot behaviour under adverse conditions. No single vital may give definitive information that a robot is failing. However, trends of a set of vitals can provide a robust indicator of adverse conditions, and may also help diagnose the nature of an adverse situation. Ideally, robot vitals should account for all potential performance degrading factors. However, we show that a small number of vitals can provide robust and rich information.

There are many different types of robots, designed and deployed for many different tasks. It is thus not easy to provide generic rules for selecting vitals metrics. However, any robot's actions should reflect its intent (e.g. its task or mission objective). Performance degradation can be defined as anything that disrupts this. The vitals are therefore a set of indicators we use to quantify how much a robot's actions are deviating from its intent. Beyond a certain threshold, this can require remediation behaviours or interventions.

We define a robot as "suffering" if it is experiencing high performance degradation. Different aspects of performance degradation are related to their corresponding vitals, by defining a probability of robot "suffering" given each vital. Hence, as the performance degradation indicated by a vital increases, the probability of suffering given the vital should increase. Each vital is calculated by sampling (and filtering) real-time data from the robot. For some vitals, non-linear transforms, such as event detection or thresholding, are applied to emphasise features of interest. Finally, the probability of suffering for each vital is calculated as a function of the resultant time series of values.

As one example realisation, we present five vitals useful for experiments with a mobile robot in our lab's mock-up disaster scenario, and the rationale in deriving them. Four vitals capture motion-related performance degradation. A fifth captures localisation-related degradation. Probability distributions and transforms for each vital are determined empirically, based on preliminary experiments, observations of multiple robot platforms, and our previous work on variable autonomy robots [1], [2], [12].

A. Rate of Change of Distance From Navigational Goal ($\dot{d}_g$)

This vital is used to indicate situations in which performance degrading factors cause a robot to not move towards its navigational goal. Such situations can be detected by observing the Rate of Change (RoC) of distance from a robot's current position to its current navigational goal ($\dot{d}_g$). The odometry position estimate obtained after Extended Kalman Filter sensor fusion is used as the robot's current position, and the goal is given by an operator or the navigation algorithm. The d_g is calculated using Euclidean distance to make minimum assumptions about the task, the algorithm used, and whether the map is known before the task. However, for more sophisticated applications d_g can be calculated using the distance remaining along a non-linear path. During little to no performance degradation (i.e., ideal behaviour), the robot moves towards the goal with uniform velocity. This results in a constant value $\dot{d}_g < 0$, barring few fluctuations. A $\dot{d}_g \approx 0$ has very little similarity to ideal behaviour and indicates that a robot is unable to move. Lastly, $\dot{d}_g > 0$ is dissimilar to ideal behaviour, and indicates that performance degradation has resulted in the robot taking a sub-optimal path or moving away from the goal. To calculate the magnitude of similarity (d_{event}), we observe $\dot{d}_g$ values over multiple time steps, and compare it to ideal behaviour using a convolutional matched filter [24]. Preliminary experiments showed that $d_{event} > 0.3$ indicates a high degree of similarity i.e., the robot is facing little to no performance degradation. A $d_{event} < -0.3$ indicates dissimilarity and suggests that the robot is unable to move or is moving away from the goal. The probability of suffering is calculated as a function of d_{event} such that its value is high if $d_{event} < -0.3$ and low if $d_{event} > 0.3$. To increase the sensitivity of the probability distribution to $d_{event} \in [-1, 1]$, we use the sigmoid function given below with constants $a = -6$ and $b = -0.15$:

$$P(\textit{suffering}|\dot{d}_g, \dot{d}_{event}) = \frac{1}{1 + \exp\left((-a \cdot d_{event} + a \cdot b)\right)} \tag{1}$$

B. Jerk Along Axis of Motion ($\dot{a}_z$)

This vital detects situations in which performance degrading factors like uneven terrain may result in sudden jerks or jittering along the axis of motion (z axis generally). Sudden dips in terrain elevation can rapidly increase the force on one side, thereby causing the robot to tilt or topple. A higher magnitude of jerk indicates that the robot is more likely to topple, making the probability of suffering higher. During preliminary experiments with a simulated Clearpath Husky robot, we observed that sudden jerks of $\pm$ 30 degrees ($\approx \pm 0.5$ radians) or above along the z-axis may increase the likelihood of the robot toppling over. Therefore, the probability of suffering given jerk along the Z-axis should be high when $\mid \dot{a}_z \mid \approx \pm 0.5$ radians, and low if $\mid \dot{a}_z \mid \approx \pm 0$.

The jerk magnitude along the axis of motion is calculated using the rate of change of linear acceleration along the Z axis $\dot{a}_z$. a_z is usually measured using an Inertial measurement unit (IMU). Since IMU readings tend to be noisy, raw IMU output values smoothened using a rolling window average and then downsampled to one reading per second before calculating $\dot{a}_z$. The function for $P(\textit{suffering}|\dot{a}_z)$ is calculated using an inverted bell curve as follows:

$$P(\textit{suffering}|\dot{a}_z) = 1 \; - \; \frac{1}{(2\pi)^{\frac{1}{2}}\,\sigma_1} e^{\left(-\left(\frac{0.5}{\sigma_2^2}(\dot{a}_z)^2\right)\right)} \tag{2}$$

The values of $\sigma 1$ and $\sigma 2$ were calculated as 0.4 and -0.9 respectively to get the probability of failure close to 1 as $\dot{a}_z$ gets close to ± 0.5.

C. RoC of Localisation Error ($\dot{\delta}_{loc}$)

Robots sometimes encounter situations where its wheels are free to rotate, but the robot itself is stuck. Uneven terrain is an example of one such performance degrading factor. As the wheels continue spinning, the robots raw odometry estimate(x_1) continues to change. However, other position estimates from visual odometry [25] or EKF sensor fusion (x_2) remain relatively constant. Such situations result in localisation errors ($\delta_{loc} = x_1 - x_2$), i.e., the difference between redundant position estimates [17] of a robot. Different SLAM algorithms are robust to different levels of localisation errors (δ_{loc}). However, the performance of a robot deteriorates after prolonged periods of high δ_{loc}. Hence, $\dot{\delta}_{loc}$ can be used as an indicator of when a robot's SLAM or localisation is compromised. While SLAM algorithms generally provide confidence measures, we use $\dot{\delta}_{loc}$ as a vital to reduce assumptions made about the robot's localisation algorithm.

During periods of low performance degradation, the localisation error is close to 0, barring small fluctuations. During periods of high performance degradation, the localisation error steadily increases. To detect such situations, we count the number of times steps $t_{event} = t$ that $\mid \dot{\delta}_{loc} \mid$ continuously takes a non-zero value. In preliminary experiments, we observed that robot failure became more likely when t_{event} was between 4-5 seconds. This is heuristically encoded as a function where the probability of suffering linearly increases (with scaling constant $k = 0.2$) with the value of t_{event}:

$$P(\textit{suffering}|\dot{\delta}_{loc}, t_{event} = t) = \begin{cases} k \cdot t & \text{if } t \in [0, 5], \\ 1 & \text{if } t \geq 5. \end{cases} \tag{3}$$

D. Robot Velocity ($\dot{x}$)

A robot's velocity is a salient indicator of performance degradation. During periods of low performance degradation, a robot velocity is constant unless acceleration or deceleration is required to turn, change directions, or around way points. This constant value is generally pre-set by the manufacturer or set by the operator before use. Navigation errors, SLAM algorithm limitations and hardware issues commonly cause a robot to halt during task execution, thereby causing a sharp drop in velocity. Alternatively, motor malfunctions and braking issues cause a robot to accelerate for long periods, thereby exceeding its standard operating velocity.

For this vital, the robot velocity is calculated by differentiating successive EKF fused position estimates of the robot. The probability of suffering is calculated as a function of the number of seconds ($t_{event} = t$) where a robots velocity is continuously trivial (i.e., close to 0), or exceeds the robot's max speed (1.0 m/s for a Clearpath Husky). That is, we count the number of seconds where $\dot{x} \leq \mid 0.01 \mid$ or $\dot{x} \geq \mid 1.0 \mid$. Our experiments have shown that as the value of t_{event} is generally below 3-4 seconds when the robot is facing low performance degradation. Accordingly, we encode the probability of suffering as the following sigmoid function with $a = 1.5$ and $b = 2.5$ such that the probability of suffering increases when t_{event} is higher than 3 seconds.

$$P(\textit{suffering}|\dot{x}, t_{event} = t) = \frac{1}{1 + \exp\left((-a \cdot t_{event} + a \cdot b)\right)} \tag{4}$$

E. Laser Scanner Noise Variance (σ^2_{noise})

This vital detects situations where laser scanner noise impairs a robot's ability to perceive, map, or navigate its surroundings. Noisy readings create inaccurate representations of a robot's surroundings, thereby increasing the likelihood of collisions, sub-optimal path planning, and robot failure. The methods for evaluating noise estimation or the robustness of SLAM algorithms to different types and levels of laser noise are beyond the scope of this study. Instead, we focus on the effect of additive white Gaussian noise on a Husky robot that uses the ROS navigation stack [26]. The laser scanner measurement array is first rearranged as a square gray scale image. We use the noise variance (σ^2_{noise}) of this image as an estimate of the total laser scanner noise [27]. The σ^2_{noise} value is then calculated by convolving the image with a $3x3$ mask and applying summations on the resultant matrix. Preliminary experiments with a Husky robot showed that low noise ($\sigma^2_{noise} \approx 0.7$) had little to no effect on its navigation, and as the noise increased to $\sigma^2_{noise} \approx 1.4$, the robot's likelihood of halting or failing increased. This effect of noise variance values between 0.5 to 1.5 on the robot is captured by designing $P(\textit{suffering}|\sigma^2_{noise})$ as a sigmoid function (similar to equations 1 and 4) defined over σ^2_{noise}, with constants $a = 5$ and $b = 1$.

F. Applying the Robot Vitals Framework to Different Cases

The above-presented vitals should work well for most types of mobile robot navigation scenarios, provided that the effects of the environment on performance degradation are captured by one or more vitals. However, some parameters will need tuning for different platforms in different settings. Such tuning can done empirically or learned from simulations or real robot experiments, and/or inferred from specifications of the robot's sensor systems.

Consider the case of adapting Robot Vitals from our rugged Clearpath Husky robot to a small lower-spec Turtlebot. After pre-processing robot data to create the vitals, the thresholds for event detection need to be adjusted. For example, the jerk ($\dot{a}_z$) required to topple a Turtlebot may be less than for the Husky. Alternatively, consider the case of a high-spec robot, with very advanced SLAM system designed for operating in noisy, uncertain environments. In this case, the minimum threshold of (σ^2_{noise}) to indicate performance degradation may need to be set higher, and can be determined through simulations with varying noise levels. Finally, once the event thresholds are fixed, the $P(\textit{suffering})$ function's shape can be modified by adjusting its constants.

Overall, the vitals-health paradigm should, in principle, apply more widely to other types of tasks and robots, e.g., manipulation scenarios, UAVs, or underwater robots. Selecting and tuning new vitals for new applications hinges on the questions: "What are the key ways in which the robot can fail?". "What are the sensor readings when performance starts to degrade, and what processing, functions (and function parameters) of the sensor readings robustly detect this?". Ultimately, a robot vital translates to a probabilistic model of a robot suffering given vital (with or without transforms applied to emphasise of interest), $P(\textit{suffering}|v)$.

IV. Robot Health

The robot health is an overall scalar estimate of a robot's ability to carry out its tasks without its capabilities being impaired by any performance degrading factors. The health can be monitored to detect situations where operator intervention or correction actions are required to improve a robots performance or prevent an imminent failure.

Robot health combines the effect of several performance degrading factors into a single meta-metric. This is somewhat analogous to expert ensembles [28], or strategic decision-making that draws on opinions from diverse subject experts in crisis management. Exploring the merits and demerits of different "combination of experts" AI methods (e.g., sum, product, context aware weighted averages) is beyond the scope of this letter. Here, for proof of principle, we compute robot health in terms of probability of robot "suffering" given the robot vitals (see section III). The total probability of robot suffering at time t is:

$$P(\textit{suffering})\big|_t = \eta \sum_{v \in V^t} P(\textit{suffering}|v)P(v)|_t \tag{5}$$

Where v is any robot vital from $V = \{\dot{d}_g, \dot{a}_z, \dot{\delta}_{loc}, \dot{x}, \sigma^2_{noise}\}$ at time t, and η is a normalisation constant. For the sake of simplicity, the probability of observing each vital $P(v)$ is assumed to be 1, i.e., a perfect observation model. In future work, this could be replaced by sophisticated models (e.g., machine learning based) that use context-aware weightings, incorporating factors such as communication delays or component wear and tear.

Information entropy is a standardised metric used to quantify the amount of information uncertainty or 'surprise' in a random variable's possible outcomes. Low entropy is observed when a robot is operating under little to no performance degradation (i.e. 'normal' operating conditions). Sudden severe performance degradation, or a gradual rise in degradation, will cause entropy to increase. If the problems are mitigated and the robot returns to normal operating conditions, then entropy will fall again. To associate high and low health with low and high performance

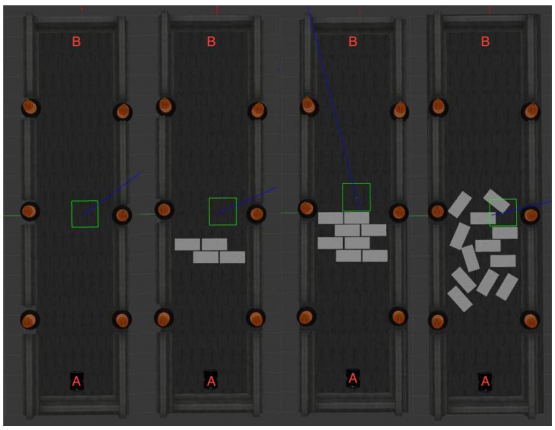

Fig. 1. The simulated arenas: From left to right, uneven terrain covers 0%, 10%, 20%, and 40% of the area respectively.

TABLE I
DIFFERENT CONDITIONS TESTED IN THE EXPERIMENTS

Experiment I, 15 Trials in each condition				
	Intensity of Performance Degradation			
	Baseline	Level 1	Level 2	Level 3
Noise Scale	0	0	0	0
% Uneven Terrain	0	10	20	40
	Level 1	Level 2	Level 3	
Noise Scale	0.4	0.5	0.6	
% Uneven Terrain	0	0	0	
	Level 1	Level 2	Level 3	
Noise Scale	0.4	0.5	0.6	
% Uneven Terrain	10	20	40	
Experiment II, 14 Trials in each condition				
	Intensity of Performance Degradation			
	Level 1	Level 2	Level 3	
Noise Scale	0	0	0.4	
HF Terrain Used	No	Yes	Yes	

degradation respectively, we use the additive inverse of information entropy as the robot health. The robot health between two time intervals t_1 and t_2 is calculated using information entropy as follows:

$$H^{t_1:t_2} = \sum_{t=t_1}^{t=t_2} P(suffering)\big|_t \cdot log(P(suffering)\big|_t) \quad (6)$$

V. EXPERIMENTAL VALIDATION

We validate our approach with experiments, in which we introduce different levels and types of performance degradation during an autonomous mobile robot navigation task. We observe corresponding effects on the robot's health and overall task performance. Two experiments were carried out using a Husky Robot, the first in simulation (Exp. I) and the second using a real robot (Exp. II). In each experiment, the robot is tasked with autonomously navigating from point A to B in the arenas shown in Figs. 1 and 3. A repository containing the ROS code for robot vitals and robot health, and all code necessary to replicate our experiments, is provided under MIT license.[1]

In both experiments, the robot uses the ROS navigation stack [26] with the dynamic window local planner and a global planner using Djikstra's algorithm. Rotate recovery behaviour was disabled to minimise confounding factors. The robot has no prior information about the map, terrain, boundary conditions or any performance degrading factors. If a robot aborts its navigation plan due to terrain adversities, laser scanner noise or a path planning timeout, the navigational goal is reset. However, if the robot is stuck, unable to find a path to the goal or aborts navigation for 30 seconds, despite resetting the goal, the experimental trial is terminated.

Three common field repairable performance degrading factors observed in adverse environments were used in the experiments: high friction (HF) terrain, uneven terrain, and laser noise. Laser scanner noise degrades the robot's ability to perceive its environment, introducing localisation and navigation errors. Such errors influence the values of vitals $\dot{d}_g$, $\dot{x}$, and σ^2_{noise}. Traversing uneven terrain causes instability. Changes in elevation or inclination may result in the laser scanner detecting the ground as an obstacle, thereby degrading navigation. Such errors affect the values of $\dot{d}_g$, $\dot{x}$, $\dot{a}_z$, and $\dot{\delta}_{loc}$. Finally, robots face difficulty turning and moving smoothly on HF terrain. Robots may slip, skid, or even halt on such surfaces. The values of $\dot{d}_g$, $\dot{x}$, and $\dot{\delta}_{loc}$ are affected in such cases. Different intensities of these performance degrading factors were combined to create multiple experimental conditions (see Table. I).

A total of 15 trials were carried out for each degradation level in Exp I and 14 trials in Exp II. After each experimental trial, we measured the time to task completion T_{comp}. Additionally, we extracted the robot health measured during runtime, and calculated average robot health for each experimental trial. Performance degradation increases the time taken by a robot to complete its task. Hence, we use T_{comp} as an objective post-hoc performance metric. By increasing the performance degradation factors in the task, we hypothesise that: 1) The value of T_{comp} increases, 2) the average robot health over the experiment runtime decreases and, 3) the average robot health and the T_{comp} are inversely correlated.

A. Experiment I

Gazebo, a high-fidelity robotics simulation with a realistic physics engine, was used for Exp I. The robot was equipped with wheel encoders for odometry, an LMS-111 LIDAR scanner and a UM6 IMU sensor. Seven seconds after the start of each trial, varying degrees of random additive Gaussian white noise were introduced into the laser scanner to degrade the robot's localisation using the Box-Muller transformation [29]. Laser noise was turned off after a further seven seconds. The duration of noise was chosen heuristically, such that substantial performance degradation was induced during runtime, without causing full robot failure. To control noise magnitude, the standard deviation of the Gaussian kernel is multiplied by several noise scale values, Fig. 1. For Exp I, we also designed four different arenas with uneven terrain as shown in Fig. 1. These arenas had

[1]https://github.com/anikethramesh/robotVitals

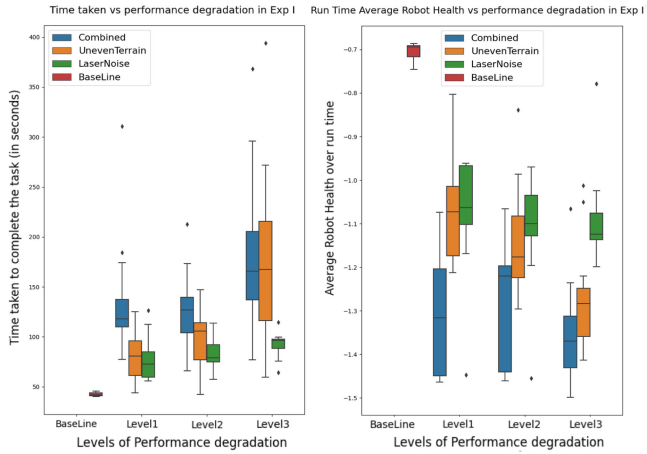

Fig. 2. Exp I: **Left:** T_{comp} (lower is better); **Right:** Average robot health (higher is better). The diamonds represent outliers.

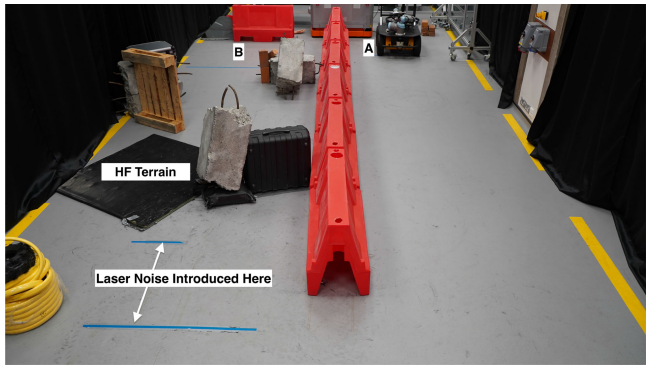

Fig. 3. The arena in Exp II: Points A and B, marked with blue tape, denote the start and goal positions. The different performance degrading factors used are annotated.

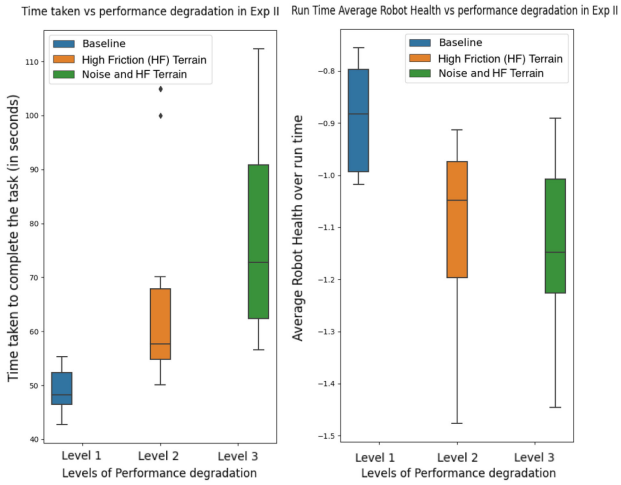

Fig. 4. T_{comp} (left) and average robot health (right) boxplots for Exp II. The diamonds represent outliers.

0%, 10%, 20% and 40% of their total area covered with the "FRC 2016 Rough Terrain" gazebo block.

Results of Experiment I: The T_{comp} and average health values for each trial, sorted by the different levels of performance degradation, are plotted in Fig. 2. With no performance degrading factors, the average value of T_{comp} is 40 seconds and average robot health during runtime is -0.7. Performance degradation increases T_{comp} and reduces average health. However, the ranges of T_{comp} and average health for different levels of performance degradation vary. The combination of laser noise and uneven terrain results in more performance degradation than either of these factors alone. This evidence suggests that the effect of multiple performance degrading factors on robots may not always be additive in nature, and that quantising the effects of performance degrading factors into different levels is complex and often non-linear. However, their intensities can be measured in relative terms. Among different performance degrading factors, laser noise induces the lowest level of degradation (minimum increase in T_{comp} from baseline performance). The presence of noise and uneven terrain results in the lowest range of average health values (-1.22 to -1.484). Finally, a Spearman's rank correlation test showed a strong and significant negative correlation ($p < 0.001$, $\rho = -0.93$) between T_{comp} and the average robot health.

B. Experiment II

Exp II was carried out using a real Husky robot in the experimental setup depicted in Fig. 3. Our robot does not possess an IMU, hence $\dot{a}_z$ values were not calculated. Robot health was calculated using $V = \{\dot{d}_g, \dot{\delta}_{loc}, \dot{x}, \sigma^2_{noise}\}$ on (5). To avoid risking hardware damage from the robot tipping over, uneven terrain was not used for the real robot experiments. Instead, HF terrain and obstacles were used to degrade the ability of the robot to turn and move smoothly. A square tile wrapped with a high friction rubber mat was used as HF terrain for this experiment. This tile was not fixed on the floor, as slippage of the tile during robot turns induces odometry and localisation errors thus adding to performance degradation. The arena used for Exp II, and the position of the HF terrain is shown in Fig. 3. Artificial laser noise was introduced in the annotated area for a period of 5 seconds. The locations where performance degradation was introduced were marked and kept constant.

Results of Experiment II: Average health and T_{comp} values for different levels of performance degradation are plotted in Fig. 4. The range of T_{comp} values increases with levels of performance degradation. The middle quartile of average health observed in level 3 is lower than that of level 2, however the range of values are similar. The robot was unable to complete the task in 2 trials of level 2, and 3 trials of level 3. In these trials the robot momentarily experienced a localisation error on the HF terrain, and then could not find a collision-free path, even after resetting the goal. The Spearman's Rank Correlation test showed a significant ($p < 0.001$), with strong negative correlation ($\rho = -0.77$) between the average robot health and T_{comp} values.

The variation of instantaneous robot health values over time, for various different combinations of degradation factors, is plotted in Fig. 5. Given the same experimental conditions, robotic

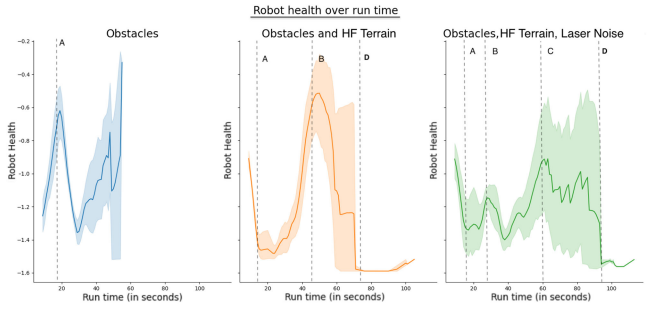

Fig. 5. The health trend over runtime for all conditions in Exp II. (L to R) Level 1, Level 2, Level 3. Error bands around the lines represent the 95% confidence interval of values. Dotted lines indicate when the robot entered the area with obstacles (A); HF terrain (B); laser noise (C); (D) indicates the approximate timestamp where robot failures were observed.

hardware and navigation algorithms, the performance degradation induced in a robot varied in each trial. This variation is due to combination of a stochastic path-planner, with stochastic noise, and physical terrains that may cause different outcomes for small variations in e.g. approach angles.

As seen in Fig. 5, the health trend for level 1 dips during the 20 to 30 seconds time period. In this period, the robot encountered obstacles, and slowed down to create a new navigation plan. The health then continued to rise until task completion, indicating little further performance degradation. In level 2, the health sharply drops around the 50 s mark. This is consistent with the robot encountering HF terrain and facing navigation errors. In level 3, the robot health trend was characterised by high fluctuations due to the combined effect of laser noise, obstacles and HF terrain. The introduction of multiple performance degradation factors in quick succession causes the health to stay below -0.8 throughout the runtime.

VI. Discussion

Our experimental results illustrate how, as the severities and types of performance degrading factors increase, T_{comp} increases (indicating posthoc, an overall decline in mission performance). Meanwhile the average robot health also decreases (see Fig. 2 and 4). This suggests that our proposed health metric, which combines inputs from our proposed vitals metrics, provides a useful indication of the degree to which a robot suffers difficulties.

Additionally, robot health values, at each time step, successfully track robot performance degradation in runtime, Fig. 5. Furthermore, there is a strong negative correlation between T_{comp} and average robot health. This suggests that using our proposed system, a robot can estimate its own performance degradation online during missions, and that it has similar accuracy as commonly used offline, post-hoc performance metrics, i.e., T_{comp}.

Hence, our proposed system of "robot vitals" and "robot health" can successfully provide an autonomous robot with awareness about its own performance degradation, at any instant, while executing missions. The same approach can also be applied to semi-autonomous (shared control or variable autonomy) systems in which a human and autonomous agents collaborate to control a remote robot. This kind of self-monitoring is an important step towards avoiding catastrophic mission failures, since it can detect deteriorating performance, and trigger remediation measures such as remote human intervention and supervision, or autonomous recovery behaviours.

Exp II provides insights on the utility of such a system. In some trials, the robot got stuck momentarily, or failed when it was unable to find a collision free path through the arena. During these instances the robot's health dropped below -1.4. Thus, a threshold-based control switcher [2] could be used to initiate recovery behaviours. Alternatively, an operator could take control of the robot to provide it a collision free path, or teleoperate it. Furthermore, in "one human, many robots" multi-robot paradigms, robot health of each robot can be used to prioritise those robots most in need of operator attention.

In a variety of applications, but especially in high-consequence or extreme environment applications, there is increasingly a demand for "explainable AI". The vitals-based health metric provides intuitive explainability, with the different vitals providing rich diagnostics information to help humans understand any particular problematic circumstance. While comparatively less explainable and intuitive, a set of vitals can also be found using methods like principal component analysis or machine learning. Such approaches can mine large amounts of data from robot operation to find parameters that best represent its performance degradation.

Lastly, while the health metric in Exp I used five vitals, Exp II used only four, as the robot lacked an IMU. This illustrates the generalisability and scalability of our framework, which is intended to handle different numbers and types of vitals according to different robots and tasks. Additional vitals can be readily added to the health metric using a $P(suffering|v)$ function that relates changes in the vital with the probability of the robot suffering. This function can be derived from Monte Carlo studies, reliability studies, or expert knowledge.

VII. Conclusion

This letter has proposed a framework of robot vitals and robot health, enabling a robot to detect and quantify its own performance degradation online, for each time-step, during task execution. We outline a systematic approach for designing vitals to suit different robots, tasks and environments, and a scalable framework for combining information from multiple vitals into a single overall health metric.

We present experiments where both simulated and real mobile robots encounter performance degradation of various types and severities. Results suggest that our framework can detect the presence and severity of performance degradation, caused by a wide variety of circumstances. Instantaneous online detection of performance deterioration is demonstrated, which correlates strongly with time stamps at which different adversities are encountered. Scalability and robustness to addition or removal of individual vitals is evidenced.

In future work, we aim to: i) evaluate the utility of this framework in assisting human-initiative, robot-initiative, and

mixed-initiative paradigms in variable autonomy systems [1], [2], [12]; ii) explore how machine learning, and statistical analysis methods, can be used to automatically extract vitals, learn optimal weights for each vital, or replace a hand-coded health function with a learned optimal fusion of vitals data; iii) extend these ideas to multi-robot systems.

REFERENCES

[1] M. Chiou, R. Stolkin, G. Bieksaite, N. Hawes, K. L. Shapiro, and T. S. Harrison, "Experimental analysis of a variable autonomy framework for controlling a remotely operating mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 3581–3588.

[2] M. Chiou, N. Hawes, and R. Stolkin, "Mixed-initiative variable autonomy for remotely operated mobile robots," *ACM Trans. Hum.-Robot Interact.*, vol. 10, no. 4, pp. 1–34, 2021.

[3] C. M. Humphrey, C. Henk, G. Sewell, B. W. Williams, and J. A. Adams, "Assessing the scalability of a multiple robot interface," in *Proc. 2nd ACM/IEEE Int. Conf. Hum.-Robot Interact.*, 2007, pp. 239–246.

[4] R. R. Murphy, "Navigational and mission usability in rescue robots," *J. Robot. Soc. Jpn.*, vol. 28, no. 2, pp. 142–146, 2010.

[5] G. B. Smith et al., "The national early warning score 2 (news2)," *Clin. Med.*, vol. 19, no. 3, pp. 260–260, 2019.

[6] S. Jiang and R. C. Arkin, "Mixed-initiative human-robot interaction: Definition, taxonomy, and survey," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2015, pp. 954–961.

[7] A. Kolling, P. Walker, N. Chakraborty, K. Sycara, and M. Lewis, "Human interaction with robot swarms: A survey," *IEEE Trans. Hum.-Mach. Syst.*, vol. 46, no. 1, pp. 9–26, Feb. 2016.

[8] J. Carlson and R. R. Murphy, "How UGVs physically fail in the field," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 423–437, Jun. 2005.

[9] P. H. Tsarouhas and G. K. Fourlas, "Mission reliability estimation of mobile robot system," *Int. J. Syst. Assurance Eng. Manage.*, vol. 7, no. 2, pp. 220–228, 2016.

[10] S. Honig and T. Oron-Gilad, "Understanding and resolving failures in human-robot interaction: Literature review and model development," *Front. Psychol.*, vol. 9, 2018, Art. no. 861.

[11] G. Steinbauer, "A survey about faults of robots used in robocup," in *Robot Soccer World Cup*. Berlin, Germany: Springer, 2012, pp. 344–355.

[12] M. Chiou, N. Hawes, R. Stolkin, K. L. Shapiro, J. R. Kerlin, and A. Clouter, "Towards the principled study of variable autonomy in mobile robots," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, 2015, pp. 1053–1059.

[13] E. Khalastchi and M. Kalech, "On fault detection and diagnosis in robotic systems," *ACM Comput. Surv.*, vol. 51, no. 1, pp. 1–24, 2018.

[14] D. J. Brooks, "A human-centric approach to autonomous robot failures," Ph.D. dissertation, Univ. Massachusetts Lowell, Lowell, MA, USA, 2017.

[15] C. G. M. Garza, "Failure is an option: How the severity of robot errors affects human-robot interaction," M.S. thesis, Carnegie Mellon Univ., Pittsburgh, PA, USA, 2018.

[16] L. Yu, M. Wu, Z. Cai, and Y. Cao, "A particle filter and SVM integration framework for fault-proneness prediction in robot dead reckoning system," *WSEAS Trans. Syst.*, vol. 10, no. 11, pp. 363–375, 2011.

[17] J. P. Mendoza, M. M. Veloso, and R. Simmons, *Mobile Robot Fault Detection Based on Redundant Information Statistics*. Pittsburgh, PA, USA: Carnegie Mellon Univ., Jun. 2018, doi: 10.1184/R1/6607376.v1.

[18] R. Wegner and J. Anderson, "Agent-based support for balancing teleoperation and autonomy in urban search and rescue," *Int. J. Robot. Automat.*, vol. 21, no. 2, pp. 120–128, 2006.

[19] A. Valero, M. Mecella, F. Matia, and D. Nardi, "Adaptative human-robot interaction for mobile robots," in *Proc. 17th IEEE Int. Symp. Robot Hum. Interactive Commun.*, 2008, pp. 243–248.

[20] B. Doroodgar, Y. Liu, and G. Nejat, "A learning-based semi-autonomous controller for robotic exploration of unknown disaster scenes while searching for victims," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2719–2732, Dec. 2014.

[21] A. Hong, O. Igharoro, Y. Liu, F. Niroui, G. Nejat, and B. Benhabib, "Investigating human-robot teams for learning-based semi-autonomous control in urban search and rescue environments," *J. Intell. Robot. Syst.*, vol. 94, no. 3/4, pp. 669–686, 2019.

[22] V. Klingspor, K. J. Morik, and A. D. Rieger, "Learning concepts from sensor data of a mobile robot," *Mach. Learn.*, vol. 23, no. 2, pp. 305–332, 1996. [Online]. Available: http://archive.ics.uci.edu/ml

[23] A. Ramesh, M. Chiou, and R. Stolkin, "Robot vitals and robot health: An intuitive approach to quantifying and communicating predicted robot performance degradation in human-robot teams," in *Proc. Companion ACM/IEEE Int. Conf. Hum.-Robot Interact.*, 2021, pp. 303–307.

[24] G. Turin, "An introduction to matched filters," *IRE Trans. Inf. Theory*, vol. 6, no. 3, pp. 311–329, Jun. 1960.

[25] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry for ground vehicle applications," *J. Field Robot.*, vol. 23, no. 1, pp. 3–20, 2006.

[26] E. Marder-Eppstein, E. Berger, T. Foote, B. Gerkey, and K. Konolige, "The office marathon: Robust navigation in an indoor office environment," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 300–307.

[27] J. Immerkaer, "Fast noise variance estimation," *Comput. Vis. Image Understanding*, vol. 64, no. 2, pp. 300–302, 1996.

[28] T. G. Dietterich et al., "Ensemble learning," *Handbook Brain Theory Neural Netw.*, vol. 2, no. 1, pp. 110–125, 2002.

[29] G. E. Box, "A note on the generation of random normal deviates," *Ann. Math. Statist.*, vol. 29, pp. 610–611, 1958.