Comparative Assessment of Sensing Modalities on Manipulation Failure Detection

Arda Inceoglu and Gökhan Ince and Yusuf Yaslan and Sanem Sariel

***Abstract*— Execution monitoring is important for the robot to safely interact with its environment, and to successfully complete the given tasks. This is because several unexpected outcomes that may occur during manipulation in unstructured environments (i.e., in homes) such as sensory noises, improper action parameters, hardware limitations or external factors. The execution monitoring process should be continuous for effective failure detection and prevention if possible. We present an empirical analysis of proprioception, audition and vision modalities to detect failures on a selected tabletop object manipulation actions. We model failure detection as a binary classification problem, where the classifier uses high level predicates extracted from raw sensory measurements. We evaluate the contributions of these modalities in detecting failures for pick, place and push actions on a Baxter robot.**

I. INTRODUCTION

Safety of industrial robots in engineered environments is a well-studied topic which is regulated with established standards [1], [2]. However, safe task execution for robots operating in unstructured environments such as kitchens remains an open issue. A robot may fail while manipulating an object resulting in undesired consequences. Sensor/motor misalignments, dropping the object due to an unstable grasp, collusions with other objects while carrying an object due to perception errors can be given as example root causes for failures. A sample failure situation is presented in Figure 1 where a Baxter robot grasps the cereal box from a wrong orientation, therefore, it produces an unstable grasp resulting in the dropping of the box while carrying.

In order to ensure the safety of the robot itself and the environment, the robot's task execution should be continuously monitored. Therefore, a continual execution monitoring and failure detection system is needed to detect anomalies in an observed state. In this study, we analyze the continuous observation data produced by various sensor modalities on a selected set of manipulation actions and their suitability for detecting failures on these actions. Our analysis includes data from proprioceptive, auditory and visual sensors for their use as past experiences to learn success and failure models. We first analyze outputs from each sensor modality separately and show that each has a different contribution in reliably detecting different anomalies. To the best of our knowledge, this is the first time that different sensor modalities are analyzed for detecting manipulation failures in such a low-level/granularity. We show how these modalities can complement each other for detecting failures for pick, place and push actions. We believe this analysis is useful for developing an effective failure detection and safe execution framework.

This research is funded by a grant from the Scientific and Technological Research Council of Turkey (TUBITAK), Grant No. 115E-368.

Authors are with Faculty of Computer and Informatics Engineering, Istanbul Technical University, Maslak, Turkey {inceoglua, gokhan.ince, yyaslan, sariel}@itu.edu.tr

Fig. 1: The Baxter robot manipulating the cereal box. An unstable grasp is produced due to a wrong grasping orientation. The box may drop during movement.

II. RELATED WORK

There is not a detailed analysis of modalities for manipulation failure detection in literature. However, it would be relevant to review existing execution monitoring and single/multi-modal failure detection methods. Detailed surveys on execution monitoring approaches can be found in [3] and [4].

Execution monitoring approaches can be grouped into [5]: (i) model-based approaches using models to predict outcomes of actions where predictions are compared with observations, and (ii) model-free approaches which are based only on observations without existing models.

Among model-based approaches [6], [7], [8] address fault detection for mobile robots. [6] uses odometry information to detect and identify differential drive faults. [7] proposes a Kalman Filter (KF) and Neural Network (NN) based system to detect and identify mechanical and sensor failures. Each fault type is modeled with a separate KF. An NN is trained using the residual between predictions and observations to

identify failures by selecting the relevant KF. In [8], the kinematic model is developed as the residual generator.

Temporal Logic based execution monitoring is another model-based approach. [9] uses Temporal Action Logic (TAL) to define action control formulas to achieve action monitoring for unmanned aircraft systems. [10] integrates several sensors including RGB-D camera, sonar, microphone and tactile sensors to evaluate Metric Temporal Logic (MTL) formulas defined for a mobile robot. Each sensor serves for detecting a different kind of failure.

[11] extends semantic knowledge, represented as Description Logic (DL) formulas, with probabilistic action and sensing models to cope with uncertainties. [12] addresses robot safety issues particularly for software components. They propose a domain-specific language based approach to implement safety rules to monitor software components.

In [13], anomalous regions in the state space are identified by detecting deviations from the normal execution model. In [14], after creating the plan for the given task, stochastic expectations are generated for actions. Observations are compared with expectations to detect unsatisfied conditions during runtime. In another study [15], extended action models are introduced in order to detect and recover from failures by repairing the plan.

In addition to model-based methods, model-free execution monitoring is studied using standard pattern recognition approaches [16]. [17] proposes a model-free fault detection mechanism by comparing two redundant observations from different sources. [18] proposes a sensor fusion based model-free failure detection and isolation method. Redundant sensor sets, with the same contextual information (e.g., distance), are installed on the robot. Conflicts and deviations in sensory measurements are monitored to detect failures. In order to isolate faults, a rule-based method is applied.

III. Perception Pipeline for Failure Detection

In our analysis, we consider *pick*, *place*, and *push* actions as compositions of primitive behavior sets {*move_to, approach, grasp, retreat*}, {*move_to, approach, release, retreat*} (the object is in the hand), {*move_to, approach, push, retreat*}, respectively. Each can be combined with a sensing action, {*sense*}. At the beginning of the execution, the scene is visually perceived by one of these sensing actions, and motion trajectories are generated. At the end of execution, the scene is perceived again to observe the effects of the manipulation in the environment.

A. Proprioception

The proprioception monitors the robot state; joint angles and torques measured by internal sensors. In this study, we only use the gripper status of the Baxter's two finger parallel gripper.

Proprioceptive Predicates: The gripper state is discretized into following mutually exclusive binary states by thresholding the force F and the distance between fingers D (τ_D and τ_F are distance and force thresholds):

- *Open:* Fingers of the gripper are open: $D > \tau_D$.
- *Closed:* Fingers are closed: $D < \tau_D \wedge F < \tau_F$.
- *Moving:* Fingers are either opening or closing.
- *Gripping:* Measured force exceeds a threshold, $F > \tau_F$.

B. Auditory Perception

The sound source identification system has three components: preprocessing, feature extraction and classification. The audio signal is acquired from the microphone at 16 KHz sampling rate. In the preprocessing step, the signal is divided into 32 ms frames with 10 ms step size. Each frame is transformed into the frequency domain via Fast Fourier Transform (FFT), and a Mel filterbank is applied to. The total energy of the current frame is thresholded to eliminate the background noise. The start and end points of an audio event is detected via empirically predefined onset and offset thresholds respectively. The feature vector contains the mean of the 12 Mel Frequency Cepstrum Coefficients (MFCC) of the first 10 frames after the onset, and the total duration measured between onset and offset. In the final step, a Linear Support Vector Machine is used to classify audio events.

Auditory Predicates: For relating sound data with a failure case, we identified four different events *($E = e_j$)* which are determined based on a classification procedure:

- *no event*: The lack of sound event, in which total energy of the current frame is less than the onset threshold value.
- *drop*: The sound generated after an object is dropped from any height.
- *hit*: The sound generated when the robot hits an object with its gripper.
- *ego-noise*: The sound generated by the robot.

C. Visual Perception

We use the Violet system described in [19] to create the model of the environment. The world model contains the detected objects as well as their physical properties (e.g., 3D location, size, color) and spatial predicates (e.g., on_table). The raw RGB-D data are processed with the Euclidian clustering based 3D segmentation [20] algorithm to extract object models from point clouds. The extracted point clouds are represented as bounding boxes. In some cases, more than one attached objects can be placed in a single bounding box. After creating bounding boxes, the total surface area (A) is calculated by projecting object bounding boxes (o) in the given scene (S_t) onto ground plane (i.e., xy):

$$A = \sum_{o \in S_t} o_{size}(x) o_{size}(y) \tag{1}$$

Visual Predicates: Comparing the initial and final world model states provided by visual perception, the following predicates are computed:

- ΔA: The difference in the total point cloud area (A), where the objects are spread on the table.
  $\Delta A = A_{final} - A_{initial}$
- ΔL: The difference in the observed location (L) of the target object. It is computed separately for each axis.
  $\Delta L = o^{final}_{location} - o^{initial}_{location}$

(a) (b)

Fig. 2: Object set for the (a) pick and place, (b) push actions.

IV. Analysis on Real-world Data

A. Environment Setup

We use the Baxter research robot with a two-finger electric gripper to manipulate objects placed on a table. An Asus Xtion RGBD camera and a PSEye microphone is mounted on the robot's head and the lower torso of the robot, respectively, to acquire visual and auditory data. The software system is developed using the ROS[1] and HARK[2] middleware.

B. Data Collection

During data collection, the robot is given several pick, place, and push actions and the following raw sensory data are recorded in the *rosbag* format: (i) the robot's state information (i.e., joint angles, joint torques), (ii) the raw audio signals obtained from the PSEye microphone, and (iii) the RGB and depth images obtained from the Asus Xtion pro camera.

Pick Dataset: The dataset contains observation sequences of 42 pick actions (with 21 successes and 21 failures). Task descriptions and failure causes are as follows: (i) Grasping an object lying on the table. The robot fails due to the incorrect localization of the object. (ii) Grasping an object on the top of a stack of other objects. The whole stack collapses while the robot is approaching. (iii) Grasping an object on the top of a built stack. In this case, the objects are next to each other. The failure occurs due to the wrong grasp orientation.

Place Dataset: The dataset consists of 39 place actions (i.e., 13 successes and 26 failures). The task is stacking blocks on top of each other by picking up the object from the table and putting it on top of the structure. The height of the structure varies from 2 to 4 objects. The structure collapses due to the unstable intermediate stacking.

Push Dataset: The dataset contains 32 push recordings (12 successes and 20 failures). A push action is conducted as follows: a randomly chosen object (see Figure 2 for the complete object set) is placed to a random location on the table. Then, the robot is asked to push the object in the given direction for a fixed amount of distance. The cause of the failure is faulty estimation of the contact point in most cases.

The reader should note that as manipulation trajectories are generated online using MoveIt[3], the duration of recordings may vary.

[1]http://www.ros.org/
[2]http://www.hark.jp/
[3]http://moveit.ros.org/

C. Qualitative Evaluation

Proprioception: Figure 3 presents raw measurements and corresponding discretized gripper status for four different types of pick actions. During pick dataset collection, to create anomalies an offset is added to the observed object location which resulted in three different failure patterns. In the first failure (Figure 3(a)), while the gripper approaches to the object, it hits the object resulting in its flip, but still it can grasp the object. We consider such a situation as a failure, since this causes an unsafe execution where a brittle object could easily get damaged. In the second failure (Figure 3(b)), the gripper status turns into gripping as it applies force on the object. However, the object cannot be grasped. The gripper status is updated as closed only after retreating. In the third failure (Figure 3(c)), the gripper collides with the object, and causes it to fall down. Figure 3(d) presents a success case. The similarity between (a) and (d) causes a confusion in the proprioception based failure detection. Similar observations are also made in both successful and failed place executions. In this case, after releasing the object, it is not possible to sense the object status via proprioception. In such cases, complementary modalities are helpful to correctly identifying failures from successful executions.

Audition: Audio data are informative in terms of detecting unexpected events such as dropping the manipulated object or hitting another object in the environment. This is particularly useful when the objects are out of sight or occluded. Figure 4 visualizes the waveform, spectrogram, and energy plots of drop audio event for four different objects namely: cubic block (wood), pasta box (carton, full), salt box (soft plastic, full), coffee mug (hard plastic, empty). The physical properties of objects (e.g., size, weight and material) affect the resulting audio signal. For example, the pasta box gets stable after landing the table, on the other hand coffee mug makes a rolling effect. Figure 5 presents the entire execution of a push action. In the last panel of Figure 5, classification outcomes are given at those moments sound events are detected.

The reader should note that the robot is aware of the action it is executing. This provides the flexibility to adjust action conditioned models. In our experiments, hit event is only observed during pick action due to misalignment. Therefore, it is considered only for pick action.

Vision: In Figure 6, world states for successful and failed cases are visualized where the task is stacking each object on top of the pre-build structure. In the successful case, ΔA remains unchanged, whereas it increases in the failure case as the objects are spread around. In Figure 7, the world states are visualized for the push action, where the task is pushing the object in the given distance and direction. Similarly, the ΔA remains unchanged for the successful execution, whereas it increases as the object falls down. Additionally, in the push action, the distance and direction are provided as action parameters. Therefore, this information can be useful to monitor the error on the object's displacement.

Based on our analysis, Figure 8 depicts observable action

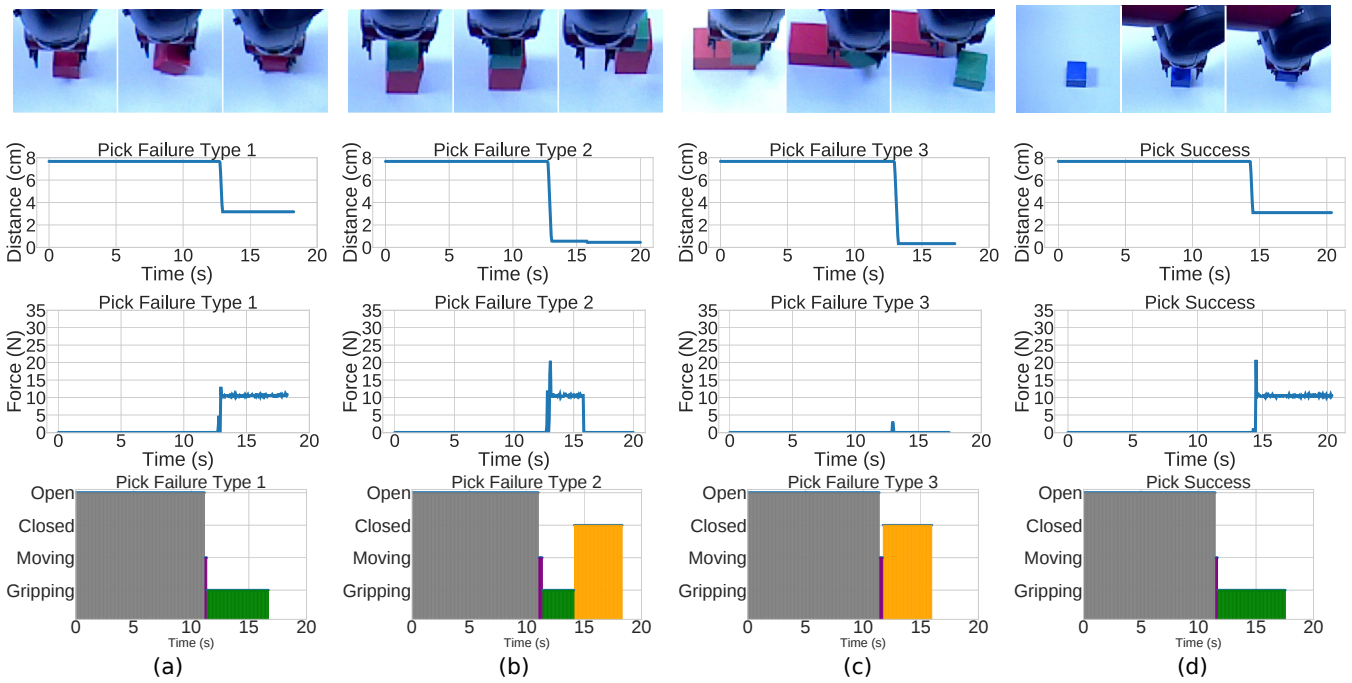

Fig. 3: The visualization of execution and measurements for four pick actions in different situations. (a-d) represent the observed data from these executions. The snapshots from the executions are presented on the top. Then, the readings on the finger distance of the two finger parallel gripper, the force on the gripper fingers and the discretized gripper status are presented correspondingly.

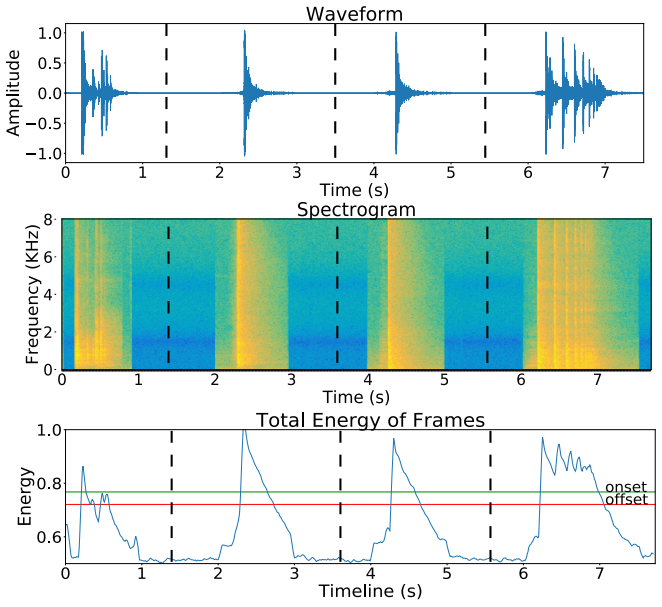

Fig. 4: Waveform, spectrogram and normalized energy plots of drop event for four different objects: cubic block, pasta box, salt box, coffee mug.

phases for each modality on the actions. Here we focus on detecting failures related to the target object to be manipulated. Proprioception data in move-to phase of pick action and retreat phase of place action does not help. In the same manner, the readings remain unchanged during the entire execution of the push action as there is no applied force in the gripper's opening/closing direction. The audio modality has no constraints and can be used to detect failures at any stage of the execution. The scene can only be observed visually before and after manipulation, where the robot arm is moved to a predefined base position such that robot arm becomes out of field of view. During the execution of the action, the robot arm partially/fully occludes the scene. Therefore, it is not feasible to make visual observations in this duration.

D. Quantitative Evaluation

The pick, place, and push datasets are randomly split into the training (50%) and the test (50%) sets by preserving their class distributions. The results are obtained by repeating the processes 10 times, and averaging them. The final decisions are made and evaluated at the end of sequences.

Hidden Markov Model (HMM) based approach is adopted to learn probabilistic temporal models from observation sequences.

1) Methods for Failure Detection:

- *Proprioception (HMM)*: A unimodal HMM based approach. An HMM is trained for each class of success and failure.
- *Audition (HMM)*: A unimodal HMM based approach on the auditory predicates. An HMM is trained for each class of success and failure.
- *Vision (ΔA)*: A prediction is made based on change in ΔA predicate. It is assumed to be failure whenever area is increased (e.g., the block tower collapses).

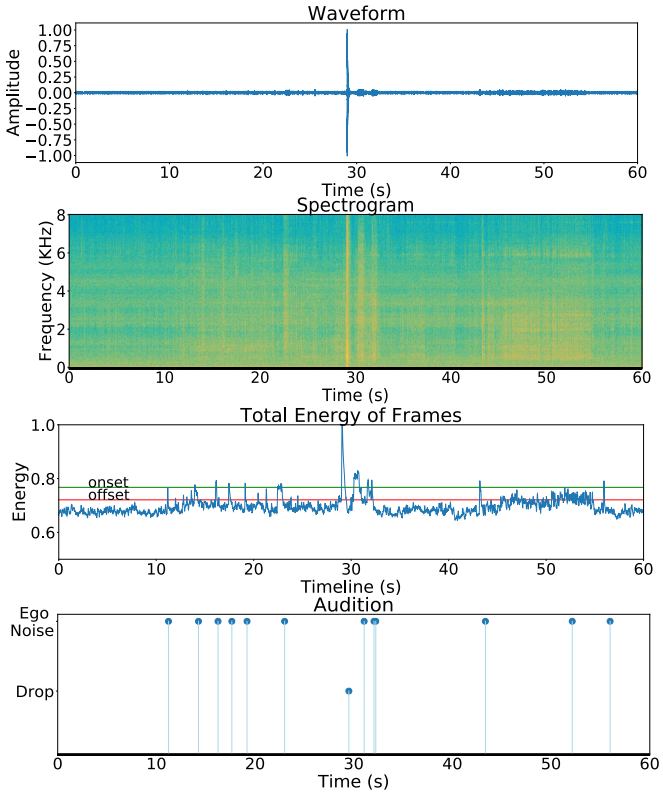

Fig. 5: Waveform, spectrogram, normalized energy plot and the classification outcome of push event for the cubic block.

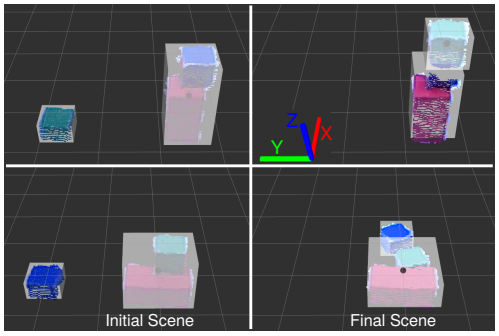

Fig. 6: Snapshots from successful (top row) and failed (bottom row) place executions. Columns correspond to initial and final scene states.

- *Vision (ΔL)*: Three binary features (i.e., for each axis) are computed using the task parameters and the difference in the observed and the expected location of the manipulated object. Then, a Decision Tree is trained.

2) Results: Table I presents the unimodal failure detection results. For the pick action, proprioception is the main source of information about the success. We are unable to assess vision performance for the pick action, due to the fact that our recordings end after grasp attempt and there is no offset to fully observe the scene.

For the place action, the proprioception is unable to provide any further feedback after releasing the object. In terms of visual representation, the difference in the total area

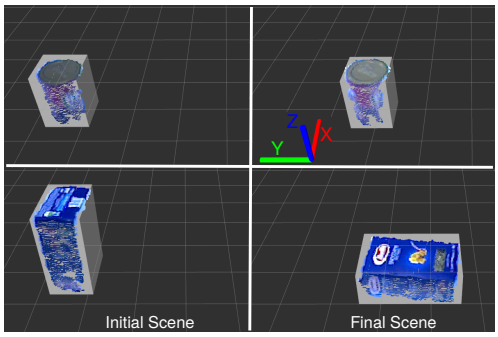

Fig. 7: Snapshots from successful (top row) and failed (bottom row) push executions. Columns correspond to initial and final scene states.

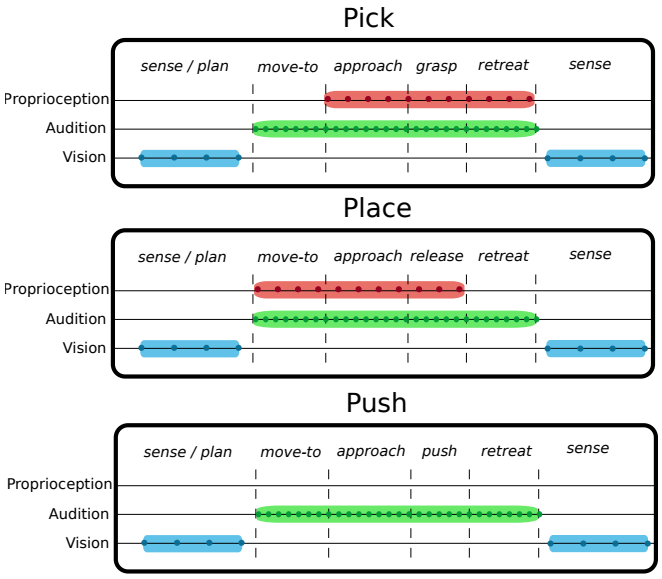

Fig. 8: Visualization of observable action phases for proprioception, audition and vision for different actions. The frequencies of the dots in rectangles roughly represent the frequencies of the readings.

can represent any major changes in the scene that results in a clutter.

During the execution of the push action, the proprioception observation remains unchanged. Object location based scene comparison performs better than area comparison, as there is only one object in the push scenario.

As can be seen in the results, proprioception modality is essential in detecting failures during the execution of pick action. This modality can be complemented by audition for some cases. For the other two actions, it is obvious that we need audition and vision, since proprioception does not provide reasonable outcomes. Comparing visual and auditory modalities, the former makes prediction with higher accuracy. However, it is needed to wait until the end of action to be able to observe the scene. On the other hand, audition can provide instant feedback.

TABLE I: Experimental Results For Unimodal Failure Detection

	Pick			Place			Push		
Approach	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**
Proprioception	0.85 ± 0.00	0.80 ± 0.00	0.80 ± 0.00	0.26 ± 0.13	0.44 ± 0.33	0.39 ± 0.10	0.48 ± 0.00	0.39 ± 0.00	0.62 ± 0.00
Audition	0.76 ± 0.05	0.76 ± 0.05	0.76 ± 0.05	0.87 ± 0.04	0.90 ± 0.02	0.87 ± 0.04	0.64 ± 0.07	0.70 ± 0.10	0.64 ± 0.07
Vision (ΔA)	N/A	N/A	N/A	0.93 ± 0.05	0.95 ± 0.03	0.93 ± 0.05	0.74 ± 0.09	0.74 ± 0.09	0.74 ± 0.09
Vision (ΔL)	N/A	N/A	N/A	N/A	N/A	N/A	0.95 ± 0.03	0.96 ± 0.02	0.95 ± 0.03

V. CONCLUSION

Execution monitoring and failure detection is a crucial component for safe autonomous manipulation in unstructured environments. In this paper, we model failure detection as a binary classification problem, and we present a failure detection system that uses semantic predicates extracted from visual, auditory and proprioceptive sensory data. We analyze when these modalities can be useful for detecting failures for picking, placing and pushing actions. As future work, we plan to create a multimodal integration framework and extend the scenarios with more daily life objects and cluttered scenes.

ACKNOWLEDGMENT

Authors also would like to thank A. Cihan Ak, B. Ongun. Kanat and Baris Bayram for their contributions in the development of manipulation and perception system.

REFERENCES

[1] ISO 10218:2011, *Robots and robotic devices – Safety requirements for industrial robots – Part 1: Robots*. ISO, Geneva, Switzerland, 2011.

[2] ISO 10218-2:2011, *Robots and robotic devices – Safety requirements for industrial robots – Part 2: Robot systems and integration*. ISO, Geneva, Switzerland, 2011.

[3] C. Fritz, "Execution monitoring – a survey," University of Toronto, Tech. Rep., 2005.

[4] O. Pettersson, "Execution monitoring in robotics: A survey," *Robotics and Autonomous Systems*, vol. 53, no. 2, pp. 73–88, 2005.

[5] J. Gertler, *Fault detection and diagnosis in engineering systems*. CRC press, 1998.

[6] D. Stavrou, D. G. Eliades, C. G. Panayiotou, and M. M. Polycarpou, "Fault detection for service mobile robots using model-based method," *Autonomous Robots*, pp. 1–12, 2015.

[7] P. Goel, G. Dedeoglu, S. I. Roumeliotis, and G. S. Sukhatme, "Fault detection and identification in a mobile robot using multiple model estimation and neural network," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, vol. 3, 2000, pp. 2302–2309.

[8] G. K. Fourlas, S. Karkanis, G. C. Karras, and K. J. Kyriakopoulos, "Model based actuator fault diagnosis for a mobile robot," in *IEEE Int. Conf. on Industrial Technology (ICIT)*, 2014, pp. 79–84.

[9] P. Doherty, J. Kvarnström, and F. Heintz, "A temporal logic-based planning and execution monitoring framework for unmanned aircraft systems," *Autonomous Agents and Multi-Agent Systems*, vol. 19, no. 3, pp. 332–377, 2009.

[10] M. Kapotoglu, C. Koc, S. Sariel, and G. Ince, "Action monitoring in cognitive robots (in turkish)," in *Signal Processing and Communications Applications Conference (SIU)*, 2014, pp. 2154–2157.

[11] A. Bouguerra, L. Karlsson, and A. Saffiotti, "Handling uncertainty in semantic-knowledge based execution monitoring," in *IROS*, 2007, pp. 437–443.

[12] S. Adam, M. Larsen, K. Jensen, and U. P. Schultz, "Towards rule-based dynamic safety monitoring for mobile robots," in *Simulation, Modeling, and Programming for Autonomous Robots*. Springer, 2014, pp. 207–218.

[13] J. P. Mendoza, M. Veloso, and R. Simmons, "Focused optimization for online detection of anomalous regions," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 3358–3363.

[14] J. P. Mendoza, M. Veloso, and Simmons, "Plan execution monitoring through detection of unmet expectations about action outcomes," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015, pp. 3247–3252.

[15] R. Micalizio, "Action failure recovery via model-based diagnosis and conformant planning," *Computational Intelligence*, vol. 29, no. 2, pp. 233–280, 2013.

[16] O. Pettersson, L. Karlsson, and A. Saffiotti, "Model-free execution monitoring in behavior-based robotics," *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 37, no. 4, pp. 890–901, 2007.

[17] J. P. Mendoza, M. M. Veloso, and R. Simmons, "Mobile robot fault detection based on redundant information statistics," 2012.

[18] A. Abid, M. T. Khan, and C. de Silva, "Fault detection in mobile robots using sensor fusion," in *Computer Science & Education (ICCSE), 2015 10th International Conference on*, 2015, pp. 8–13.

[19] A. Inceoglu, C. Koc, B. O. Kanat, M. Ersen, and S. Sariel, "Continuous visual world modeling for autonomous robot manipulation," *(to appear in) IEEE Transactions on Systems, Man, and Cybernetics: Systems*, no. 99, pp. 1–14, 2018.

[20] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze, "Point cloud library," *IEEE Robotics & Automation Magazine*, vol. 1070, no. 9932/12, 2012.