Measurement-Robust Control Barrier Functions:
Certainty in Safety with Uncertainty in State

Ryan K. Cosner, Andrew W. Singletary, Andrew J. Taylor, Tamas G. Molnar,
Katherine L. Bouman, and Aaron D. Ames

Abstract—The increasing complexity of modern robotic systems and the environments they operate in necessitates the formal consideration of safety in the presence of imperfect measurements. In this paper we propose a rigorous framework for safety-critical control of systems with erroneous state estimates. We develop this framework by leveraging Control Barrier Functions (CBFs) and unifying the method of Backup Sets for synthesizing control invariant sets with robustness requirements—the end result is the synthesis of Measurement-Robust Control Barrier Functions (MR-CBFs). This provides theoretical guarantees on safe behavior in the presence of imperfect measurements and improved robustness over standard CBF approaches. We demonstrate the efficacy of this framework both in simulation and experimentally on a Segway platform using an onboard stereo-vision camera for state estimation.

I. INTRODUCTION

Safety is of utmost importance in many modern control applications, including autonomous vehicles, medical and industrial robotics [1]. The growing complexity of these systems demands that safety properties be rigorously encoded in the controller design. Such systems are typically described as safe if their state never leaves a prescribed safe set, and Control Barrier Functions (CBFs) [2], [3] have become increasingly popular [4], [5] as a tool for achieving safety. In this paper, we focus on two challenges related to safety-critical control realized via CBFs: finding admissible inputs and making these inputs robust to uncertainty.

The first challenge is guaranteeing that a safe control input is always available. If safe control actions exist—i.e., satisfy input constraints—over the entire safe set, the set is called control invariant [6]. Yet control invariance is not guaranteed in general—safe actions may not exist for all points in a given safe set. Therefore, identifying control invariant sets is critically important for implementing safety-critical controllers in robotic systems. Hamilton-Jacobi reachability analysis can be performed to compute such sets [7], but is intractable for high dimensional systems. Here we adapt the method of Backup Sets introduced in [8] as a computationally tractable way of achieving control invariance.

II. PRELIMINARIES

First we provide a review of safety-critical control through Control Barrier Functions (CBFs) and synthesis of control
invariant sets via the Backup Set method.

A. Control Barrier Functions

Consider the nonlinear control affine system given by:

\[ \dot{x} = f(x) + g(x)u, \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m, \quad (1) \]

where \( f : \mathbb{R}^n \to \mathbb{R}^n \) and \( g : \mathbb{R}^n \to \mathbb{R}^{n \times m} \) are locally Lipschitz continuous. Given a locally Lipschitz continuous controller \( k : \mathbb{R}^n \to \mathbb{R}^m \), the closed-loop dynamics are:

\[ \dot{x} = f_t(x) = f(x) + g(x)k(x), \quad (2) \]

where \( f_t : \mathbb{R}^n \to \mathbb{R}^n \) is also locally Lipschitz continuous. Therefore, for any initial condition \( x(0) = x_0 \in \mathbb{R}^n \) there exists an interval \( I(x_0) \triangleq [0, t_{\text{max}}) \) such that \( x(t) \) is the unique solution to (2) for \( t \in I(x_0) \) [18]. Throughout this paper we assume \( I(x_0) = [0, \infty) \).

The notion of safety is formalized by defining a safe set \( C \subset \mathbb{R}^n \) in the state space that the system must remain within. In particular, consider the set \( C \) as the 0-superlevel set of a continuously differentiable function \( h : \mathbb{R}^n \to \mathbb{R} \):

\[ C \triangleq \{ x \in \mathbb{R}^n : h(x) \geq 0 \}, \]

\[ \partial C \triangleq \{ x \in \mathbb{R}^n : h(x) = 0 \}, \]

\[ \text{Int}(C) \triangleq \{ x \in \mathbb{R}^n : h(x) > 0 \}. \quad (3) \]

We assume that zero is a regular value of \( h \) and \( C \) is non-empty and has no isolated points, that is, \( h(x) = 0 \implies \partial h(x) \neq 0 \). Int\((C) \neq \emptyset \) and \( \text{Int}(C) = \overline{C} \). In this context, safety is synonymous with the forward invariance of \( C \):

**Definition 1 (Forward Invariance and Safety).** A set \( C \subset \mathbb{R}^n \) is forward invariant if for every \( x_0 \in C \), the solution to (2) satisfies \( x(t) \in C \) for all \( t \geq 0 \). The closed-loop system (2) is safe with respect to \( C \) if \( C \) is forward invariant.

We call a continuous function \( \alpha : \mathbb{R} \to \mathbb{R} \) an extended class-\( \mathcal{K}_\infty \) (class-\( \mathcal{K}_\infty \)) if it is strictly monotonically increasing and satisfies \( \alpha(0) = 0 \), \( \lim_{r \to \infty} \alpha(r) = -\infty \), and \( \lim_{r \to \infty} \alpha(r) = \infty \). Control Barrier Functions (CBF) [2] can be used to synthesize controllers which ensure the safety of the closed-loop system (2) with respect to a given set \( C \).

**Definition 2 (Control Barrier Function (CBF)).** Let \( C \subset \mathbb{R}^n \) be a safe set given by (3). The function \( h \) is a Control Barrier Function (CBF) for (1) on \( C \) if there exists \( \alpha \in \mathcal{K}_\infty \) such that for all \( x \in C \):

\[ \sup_{u \in \mathbb{R}^m} \dot{h}(x, u) \triangleq \left( \frac{\partial h}{\partial x} f(x) + \frac{\partial h}{\partial x} g(x) u \right) \geq -\alpha(h(x)), \quad (4) \]

where \( L_f h : \mathbb{R}^n \to \mathbb{R} \) and \( L_g h : \mathbb{R}^n \to \mathbb{R}^m \) are the Lie derivatives of \( h \) with respect to \( f \) and \( g \), respectively.

Intuitively, the CBF constraint (4) requires the system to slow down as it approaches the boundary of the safe set (the right-hand side of (4) increases to \( 0 \) as the value of \( h \) approaches \( 0 \)). A main result in [19], [20] relates CBFs to the safety of the closed-loop system (2) with respect to \( C \):

**Theorem 1.** Given a safe set \( C \subset \mathbb{R}^n \), if \( h \) is a CBF for (1) on \( C \), then any locally Lipschitz continuous controller \( k : \mathbb{R}^n \to \mathbb{R}^m \) satisfying:

\[ L_f h(x) + L_g h(x)k(x) \geq -\alpha(h(x)) \quad (5) \]

for all \( x \in C \), renders the system (2) safe w.r.t. \( C \).

Given a nominal (but not necessarily safe) locally Lipschitz continuous controller \( k_d : \mathbb{R}^n \to \mathbb{R}^m \) and a CBF \( h \), the CBF-Quadratic Program (CBF-QP) [2] ensures safety:

\[ \min_{u \in \mathbb{R}^m} \frac{1}{2} u^T u + \frac{1}{2} \alpha(h(x)) \quad \text{s.t.} \quad L_f h(x) + L_g h(x)u \geq -\alpha(h(x)). \]

B. Generating Control Invariant Sets via Backup Sets

To guarantee that a safe control action exists, one needs to ensure the existence of a function \( h \) satisfying the CBF condition (4). For a given safe-set \( C \), fulfilling this requirement can be nontrivial and potentially impossible. To this end, we restrict our focus to a set \( C_f \subseteq C \) which is control invariant:

**Definition 3 (Control Invariance).** A set \( C_f \subseteq C \) is control invariant if there exists a controller \( k : \mathbb{R}^n \to \mathbb{R}^m \) such that \( C_f \) is forward invariant with respect to the system (2).

While directly computing control invariant sets remains challenging in general, we may define one implicitly via a backup set [8]. Consider a desired safe set \( C \subset \mathbb{R}^n \), which is not necessarily control invariant. Suppose there exists a set \( C_B \subset C \), defined as the 0-superlevel set of a continuously differentiable function \( h_B : \mathbb{R}^n \to \mathbb{R} \), which is known a priori to be control invariant and can be rendered forward invariant by a known locally Lipschitz continuous backup controller \( k_B : \mathbb{R}^n \to \mathbb{R}^m \). We refer to \( C_B \) as the backup set. For simple backup controllers (such as linear state feedback controllers designed for the linearization of a system) it is possible to find analytical expressions for local regions of attraction to serve as backup sets. Alternatively, numerical tools such as Sums-of-Squares (SOS) may be used to synthesize control invariant sets [21].

We extend the backup set to a larger control invariant set \( C_I \subset \mathbb{R}^n \), satisfying \( C_B \subset C_I \subseteq C \), by considering the backup trajectory over a finite and fixed time \( T \in \mathbb{R}_{>0} \) as follows. By assumption, for any \( x \in \mathbb{R}^n \) there exists a unique solution \( \varphi : [0, T] \to \mathbb{R}^n \) satisfying:

\[ \frac{d}{dt} \varphi(t) = f(\varphi(t)) + g(\varphi(t))k_B(\varphi(t)) \]

\[ \varphi(0) = x. \quad (6) \]

The solution \( \varphi \) may be interpreted as the evolution of the system over the interval \([0, T]\) from a state, \( x \), under the backup controller \( k_B \). In particular, the current state \( x(t) \) may be used as initial condition in specifying \( \varphi \). We denote \( \phi_k(\tau) \triangleq \varphi(\tau) \) for the initial condition \( x \).

Using this notation, we may define the set \( C_I \subseteq C \) as:

\[ C_I = \left\{ x \in C \left| \begin{array}{l} h(\phi_k^x(x)) \geq 0, \forall \tau \in [0, T]\\ h_B(\phi_k^x(x)) \geq 0 \end{array} \right. \right\}. \quad (7) \]
The first inequality implies safety under the backup policy \((\phi^k_B(\tau)(x) \in C \text{ for all } \tau \in [0,T])\), and the second inequality implies the backup trajectory reaches \(C_B\) by time \(T\) \((\phi^k_B(\lang x \rang) \in C_B)\). The set \(C_I\) is thus control invariant as there exists at least one controller, \(k_B\), which renders it forward invariant. While \(C_I\) is not necessarily the largest control invariant subset of \(C\) (see viability kernel, [6]), the backup sets provide a computationally tractable method for finding an under-approximation of the largest control invariant set.

For notational simplicity, we define the continuously differentiable functions \(\overline{T}_\tau : \mathbb{R}^n \rightarrow \mathbb{R}\) and \(\overline{B}_\tau : \mathbb{R}^n \rightarrow \mathbb{R}\) as:

\[
\overline{T}_\tau(x) \triangleq h(\phi^k_B(\tau)(x)), \quad \overline{B}_\tau(x) \triangleq h(B(\phi^k_B(x)). \quad (8)
\]

Given these definitions, the CBF condition (4) can then be specified for the set \(C_I\) at a point \(x \in C_I\) as follows:

\[
L_T \overline{T}_\tau(x) + L_g \overline{T}_\tau(x) \geq -\alpha \overline{T}_\tau(x), \quad \forall \tau \in [0,T],
\]

\[
L_E \overline{B}_\tau(x) + L_g \overline{B}_\tau(x) \geq -\alpha \overline{B}_\tau(x). \quad (9)
\]

Any locally Lipschitz continuous controller that takes values satisfying (9) for all \(x \in C_I\) will keep the closed loop system (2) safe with respect to \(C_I\), see [22, p. 6].

We note that enforcing the first constraint in (9) is not necessarily tractable as it must hold for all \(\tau \in [0,T]\). To resolve this, it can be reduced to a finite collection of more conservative constraints through constraint tightening. A controller which implements the finite number of tightened constraints, and thus renders (2) safe with respect to \(C_I\), is given by the Backup Set Quadratic Program (BS-QP):

\[
k(x) = \arg\min_{u \in \mathbb{R}^m} \frac{1}{2} \|u - k_d(x)\|^2 \\
\text{s.t. } \quad L_T \overline{T}_{\tau_j}(x) + L_g \overline{T}_{\tau_j}(x) u \geq -\alpha(h(\tau_j)(x) - \mu),
\]

\[
L_E \overline{B}_{\tau_j}(x) + L_g \overline{B}_{\tau_j}(x) u \geq -\alpha(h(B(\tau_j)(x))
\]

for all \(\tau_j \in \{0, \Delta_t, \ldots, T\}\), where \(\Delta_t \in \mathbb{R}_{>0}\) is a time-step such that \(T/\Delta_t \in \mathbb{N}\) and \(\mu \in \mathbb{R}_{>0}\) satisfies:

\[
\mu \geq \frac{\Delta_t}{2} \max_{x \in C} \|f(x) + g(x)k_B(x)\|_2, \quad (10)
\]

with \(\Delta_t \in \mathbb{R}_{>0}\) a Lipschitz constant for \(h\) on \(C\) [8, Thm. 1].

### III. Measurement Robustness

The guarantees endowed by the above controllers require perfect knowledge of the state \(x\), which is unattainable in practice. In particular, the relationship between the state of the system and the measurements, such as images or point clouds, can be complex and not fully known [15], [16], [17]. In this section we revisit measurement-model uncertainty and present our main result in the form of a measurement-robust version of the BS-QP.

#### A. Measurement-Model Uncertainty

To achieve robustness, we consider a structured form of measurement-model uncertainty that modifies the CBF condition (4) [14]. We assume the state \(x\) is not directly available, but rather a state-dependent sensor measurement:

\[
y = p(x), \quad (11)
\]

where \(y \in \mathbb{R}^k\) and \(p : \mathbb{R}^n \rightarrow \mathbb{R}^k\) is locally Lipschitz continuous. An estimate of the state, \(\hat{x} \in \mathbb{R}^n\), is reconstructed from \(y\) (such as through measurement models or data-driven methods [16], [17]). In particular, we assume the map from measurements to state estimates is imperfect (does not recover the true state exactly), and is given by the locally Lipschitz continuous function \(\hat{q} : \mathbb{R}^k \rightarrow \mathbb{R}^n\) as follows:

\[
\hat{x} = \hat{q}(y) = x + e(x), \quad (12)
\]

where the state error function \(e : \mathbb{R}^n \rightarrow \mathbb{R}^n\) is unknown and implicitly defined by \(\hat{q}\).

The error function \(e\) can often be characterized via upper bounds on measurement-model uncertainty. In particular, we assume that while the state error \(e(x)\) is not known for a given state \(x \in \mathbb{R}^n\), it is within a compact error set \(E(x)\) specified by a set-valued function \(\hat{E} : \mathbb{R}^k \rightarrow \mathbb{P}(\mathbb{R}^n)\) (\(\mathbb{P}\) denotes the power set). That is, we have \(e(x) \in E(x)\) for all \(x \in \mathbb{R}^n\).

Since the controller only has access to the measurement and the state estimate, systems with measurement-model uncertainty evolve according to:

\[
x = f(x) + g(x)k(y, \hat{x}). \quad (14)
\]

Qualitatively this uncertainty is similar to error in the dynamics model since the true values of \(f(x)\) and \(g(x)\) are unknown to the controller. The error bound can be used to synthesize controllers which render such systems provably safe as follows [14]:

**Theorem 2.** Given a safe set \(C \subset \mathbb{R}^n\), assume that \(L_f\), \(L_g\), and \(a \circ h\) are Lipschitz continuous on \(C\) with Lipschitz constants \(\Sigma_{L_f}, \Sigma_{L_g}\), and \(\Sigma_{a\circ h}\) respectively. Define the function \(e : \mathbb{R}^k \rightarrow \mathbb{R}_{\geq 0}\) as in (13), and define the functions \(a, b : \mathbb{R}^k \rightarrow \mathbb{R}_{\geq 0}\) as \(a(y) = (\Sigma_{L_f} + \Sigma_{a\circ h})e(y)\) and \(b(y) = \Sigma_{L_g}e(y)\). If \(k : \mathbb{R}^k \times \mathbb{R}^n \rightarrow \mathbb{R}^n\) is a Lipschitz continuous controller satisfying:

\[
L_f h(x) + L_g h(\hat{x}) k(y, \hat{x}) - (a(y) + b(y)||k(y, \hat{x})||_2) \geq -\alpha(h(\hat{x})), \quad (15)
\]

for all \(x \in C\), with \(y = p(x)\) and \(\hat{x} = \hat{q}(y)\), then the system (14) is safe with respect to \(C\).

A continuously differentiable function \(h : \mathbb{R}^n \rightarrow \mathbb{R}\) for which such a controller exists is termed a Measurement-Robust Control Barrier Function (MR-CBF) [14]. As compared to the original CBF constraint (4), the MR-CBF constraint (15) adds additional terms incorporating bounds on the measurement error that ensure that the system is safe with respect to all possible states which could have generated the measurement. The original CBF constraint is recovered in the absence of measurement error, i.e. when \(e(y) = 0\).
B. Measurement-Robust Backup Set Optimization Program

In this section we present our main result in the form of a safety-critical control paradigm that is robust to measurement uncertainty. This is accomplished by unifying the Backup Set method with MR-CBFs, using the MR-CBF condition (15) the finite set of constraints imposed in the BS-QP become:

\[ L_\ell \bar{h}_{\tau_j}(\bar{x}) + L_g \bar{h}_{\tau_j}(\bar{x}) u \geq -\alpha(\bar{h}_{\tau_j}(\bar{x}) - \mu), \]

where we choose \( \nu = \mu \) if \( \bar{h} = \bar{h}_{\tau_j} \) and \( \nu = 0 \) if \( \bar{h} = \hat{B}_T \). It follows by Lipschitz continuity that:

\[
\begin{align*}
||L_\ell \bar{h}(\bar{x}) - L_\ell \bar{h}(x)||_2 &\leq \sum \alpha_i \bar{h}(i), \\
||\alpha(\bar{h}(\bar{x}) - \nu) - \alpha(\bar{h}(x) - \nu)||_2 &\leq \sum \alpha_i \bar{h}(i), \\
||L_g \bar{h}(\bar{x}) - L_g \bar{h}(x)||_2 ||k(y, \bar{x})||_2 &\leq \sum \alpha_i \bar{h}(i) ||k(y, \bar{x})||_2.
\end{align*}
\]

As \( k \) satisfies (16), we have that:

\[
c(x, k(y, \bar{x})) = c(x, k(y, \bar{x})) + c(x, k(y, \bar{x})) - c(x, \bar{x}, k(y, \bar{x})) \\
\geq c(x, \bar{x}, k(y, \bar{x})) - (a(y) + b(y)) ||k(y, \bar{x})||_2 \geq 0.
\]

Since \( c(x, k(y, \bar{x})) \geq 0 \) and \( \mu \) satisfies (10), we have that the system (14) is safe with respect to \( C_I \) by [8, Lemma 2].

This result allows us to present an alternative to the BS-QP controller which adds the measurement-robustness of MR-CBFs. The constraints (16) can be directly integrated into a Measurement-Robust Backup Set Optimization Program controller MR-BS-OP as:

\[
k(y, \bar{x}) = \arg\min_{u \in \mathbb{R}^m} \frac{1}{2} ||u - k_d(y)||^2 \quad \text{(MR-BS-OP)}
\]

\[
s.t. \quad L_\ell \bar{h}_{\tau_j}(\bar{x}) + L_g \bar{h}_{\tau_j}(\bar{x}) u \geq -\alpha(\bar{h}_{\tau_j}(\bar{x}) - \mu), \\
L_g \bar{h}_{\tau_j}(\bar{x}) + L_g \bar{h}_{\tau_j}(\bar{x}) u \geq -\alpha(\bar{h}_{\tau_j}(\bar{x}) - \mu),
\]

for all \( \tau_j \in \{0, \Delta_t, \ldots, T\} \). Since this controller is a second-order cone program (SOCOP), there exist a variety of solvers capable of implementing it including ECOS [23]. Notably, the conservative nature of the method scales with the bound on the measurement-model error \( \epsilon(y) \) and the MR-BS-OP reduces to the BS-QP when \( \epsilon(y) = 0 \). We remark that the feasibility of MR-BS-OP for all \( \bar{x} \in \mathbb{R}^n \) can be ensured by adding a slack variable to the optimization problem. The impact of the slack variable on safety can be understood via the concept of projection-to-state safety [24].
Fig. 3. Experimental results using SLAM from the onboard Intel RealSense T265 and constant desired velocity of 1 m/s. The notation and color schemes are the same as in Fig. 2. (Left) An image of the Segway platform. (Center) Trajectories generated using the BS-QP. The true trajectory exits the safe set at $t = 6.7$ s. The measurement error is plotted in blue. (Right) Trajectories generated using the MR-BS-OP. Both the true and measured trajectories are safe demonstrating the robustness of the MR-BS-OP when compared to the BS-QP.

IV. EXPERIMENTAL RESULTS

In this section we demonstrate the efficacy of the proposed MR-BS-OP controller on a modified Ninebot E+ Segway platform in both simulation and hardware experiment.

We consider a 4-dimensional asymmetrical Segway model shown in Figures 2 and 3. The state of the system consists of the position $x$, the forward velocity $\dot{x}$, the pitch angle $\psi$, and the pitch rate $\dot{\psi}$. The equations of motion were derived using Newton-Euler method treating the Segway as an inverted pendulum with control input as torque command at the wheels; see [8]. The Backup Set method for generating control invariant sets is particularly relevant for this system due to its non-minimum phase dynamics.

The desired safe set was chosen empirically to be the set of states with position less than 2 m from the origin, i.e. $C = \{x \in \mathbb{R}^n : x \leq 2\}$ and $h(x) = 2 - x$. The backup controller was an LQR controller on the linearized system dynamics and the backup set was an estimate of the region of attraction of the LQR controller to the upright equilibrium state, given by a quadratic Lyapunov function. This set is then translated to match the current position of the Segway, while not allowing it to exceed the set boundary. The functions $\overline{h}_\tau$, $\tau \in [0, T]$ were converted into four CBFs $\overline{h}_\tau$. Lastly, the Lipschitz constants for $\overline{h}_\tau$ were found explicitly by inspection of the Segway dynamics and the Lipschitz constants for $\overline{h}_B$ were found by sampling the state space in simulation and taking the largest numerical gradient.

**Simulation Results.** The MR-BS-OP was first validated in simulation in a ROS-based environment, found here\(^1\). Measurement-model uncertainty was achieved by artificially adding a constant error of $-0.4$ m to the true state. The simple test scenario involved driving the Segway forward with a constant desired velocity of 1 m/s. As seen in Figure 2, the MR-BS-OP provided robustness to this error. Importantly, without measurement-robustness, the system would be unsafe due to uncertainty in the state.

**Hardware Results.** The MR-BS-OP was then implemented on hardware. State estimates for $\dot{x}, \dot{\psi}$, and $\dot{\psi}$ were found using wheel incremental encoders and a VectorNav VN-100 IMU. The position estimate for $x$ was obtained from an Intel RealSense T265 onboard camera running proprietary Visual Inertial Odometry (VIO) based SLAM. Onboard computation was performed by a Jetson TX2 which computes control actions and relays them to the low-level motor controllers. The TX2 concurrently runs Linux with ROS, enabling external communication and logging, and the ERIKA3 real-time operating system, which enables real-time low-level communication and computation of the control action.

As the $(\dot{x}, \psi, \dot{\psi})$ state estimates provided by the encoders and IMU are highly accurate, we focus on making the system robust to measurement uncertainty in its vision-based position estimate $\hat{x}$. An OptiTrack motion capture system was used in laboratory experiments to provide $x$ estimates which are considered true. These closely matched the encoder position estimates for short trials, so the encoder $x$ estimates were considered true in the outdoor experiments. This data was used to determine the error bound $\epsilon(y)$ that appears in the MR-CBF constraint when using the onboard camera.

The value $\epsilon(y) = 0.4$ was chosen as an upper bound on the measurement error for all $y \in p(\mathcal{C})$. The MR-BS-OP was implemented at the embedded level in the ERIKA3 operating system using the ECOS SOCP solver [23]. The desired controller $k_d$ was a proportional-derivative controller tracking user velocity inputs. The backup trajectory $\phi_B^x (\hat{x})$ and its partial derivatives were approximated via Euler integration using a time step of $\Delta t = 5$ ms and the time used to expand the backup set $C_B$ to $C_I$ was $T = 1$ s. The MR-BS-OP ran at 250 Hz with 5 decision variables, 4 linear constraints, and 6 second order cone constraints and saturated at $\pm 20$ Nm.

To demonstrate the method, a simple scenario is executed on the Segway in which it is driven forward at a desired velocity of 1 m/s. This scenario is performed with both the BS-QP and the MR-BS-OP. The results of these experiments can be found in Figure 3, images from the experiment can be seen in Figure 4, and a video can be found at [25]. With the BS-QP controller the estimated state $\hat{x}$ remains safe, but the true state $x$ becomes unsafe whereas with the MR-BS-OP controller both the estimated and the true state are kept safe. This highlights the importance of providing robustness against measurement uncertainty, as achieved by Theorem 3.
Fig. 4. Images from the experiment using the MR-BS-OP controller. The Segway is piloted towards a wall of yellow boxes and the controller ensures that it remains safe, i.e. that it does not crash into the boxes. (Top) Time lapse of the Segway trajectory. (Bottom) Camera images taken from the perspective of the Segway throughout the experiment. The images are displayed in chronological order from left to right. A video can be found at [25].

V. CONCLUSION

This paper established robust controller synthesis with formal safety guarantees for systems relying on uncertain measurements. We approached this problem through the framework of CBFs. We additionally highlighted the importance of control invariant sets and experimentally implemented the Backup Set method to produce such a set for a Segway. Our theoretical construction culminated in the integration of the Backup Set method with MR-CBFs, which provides robustness to state measurement uncertainty in the safety guarantees. We implemented the proposed control method on a Segway platform and demonstrated robustly safe operation in experiments. Future work includes addressing feasibility of the MR-BS-OP for general systems and developing online methods to efficiently identify the required error bound in the context of probabilistic and time-varying uncertainties.

REFERENCES