2024 IEEE International Conference on Robotics and Automation (ICRA)
May 13-17, 2024. Yokohama, Japan

Risk-Sensitive Extended Kalman Filter

Armand Jordana, Avadesh Meduri, Etienne Arlaud, Justin Carpentier and Ludovic Righetti

2024 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-8457-4/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICRA57147.2024.10611266

***Abstract*— Designing robust algorithms in the face of estimation uncertainty is a challenging task. Indeed, controllers seldom consider estimation uncertainty and only rely on the most likely estimated state. Consequently, sudden changes in the environment or the robot's dynamics can lead to catastrophic behaviors. Leveraging recent results in risk-sensitive optimal control, this paper presents a risk-sensitive Extended Kalman Filter that can adapt its estimation to the control objective, hence allowing safe output-feedback Model Predictive Control (MPC). By taking a pessimistic estimate of the value function resulting from the MPC controller, the filter provides increased robustness to the controller in phases of uncertainty as compared to a standard Extended Kalman Filter (EKF). The filter has the same computational complexity as an EKF and can be used for real-time control. The paper evaluates the risk-sensitive behavior of the proposed filter when used in a nonlinear MPC loop on a planar drone and industrial manipulator in simulation, as well as on an external force estimation task on a real quadruped robot. These experiments demonstrate the ability of the approach to significantly improve performance in face of uncertainties.**

I. INTRODUCTION

Adapting the decisions robots make based on their perception of the world is key to deploying robots outside factories and laboratories. More precisely, controllers should adapt to the degree of certainty or confidence of the robot's belief of the world. For instance, it is important that a quadruped chooses conservative footholds and slows body movements when its confidence in the location of the ground decreases. Robust output feedback Model Predictive Control (MPC) studies methods that can adapt robot decisions based on the confidence of the perception module. However, the general nonlinear problem is very difficult, and practical algorithms remain often limited to linear systems [1].

The common practice in robotics is to decouple estimation and control, i.e., assume that the certainty equivalence principle holds [2]–[6], due to the availability of separate and tractable control and estimation algorithms. The estimation module is often a variation of a Gaussian filter, such as an Extended Kalman Filter (EKF) [7], which computes both the mean and uncertainty of the state estimates from sensor information. In control, an increasingly popular approach is MPC, which consists in solving an optimal control problem (OCP) numerically at each time step or at a fixed frequency [3], [8]–[11]. The controller can then adapt its behavior based on the current state of the robot and its environment. During deployment, an estimator is used to compute the mean of the state estimate, which is then passed on to the controller to compute the optimal behavior [2]–[6]. Unfortunately, relying on the most likely outcome can lead to catastrophic behavior. For instance, on a load-carrying task with a quadruped where the load's mass is unknown, the notion of mean might not be appropriate as this could lead the quadruped to apply insufficient force on the ground and then fall.

Some approaches address this issue by adding robustness or safety bounds in either the estimation or control block while keeping them independent. For instance, Robust Extended Kalman filtering [12] adds robustness to inaccuracies of the EKF or the model. However, the control objective is disregarded, and thus the controller cannot be robust to estimation uncertainties. Robust MPC has been studied and applied to robots, e.g. to control a biped robot using tube-based MPC [13] or linear stochastic MPC [14]. However, this line of work assumes the state to be known. In contrast to such approaches, we aim to link estimation and control by adding into the estimation module a notion of control performance to improve robustness to the estimation uncertainty. While robust output-feedback MPC controllers have been investigated [15], [16], they have not been deployed on robots due to their high computational cost. In [17], we proposed an efficient algorithm to solve the dynamic game control with imperfect state observation formulation introduced by [18].

In this paper, we leverage this result to introduce the Risk-Sensitive Extended Kalman Filter (RS-EKF), a novel filter that enables online risk-sensitive output feedback MPC at a low computational cost. The RS-EKF computes state estimates robust to measurement uncertainty while taking into account the value function provided by the controller, i.e., the estimator tailors risk reduction to the control objectives. This, in turn enables automatic modification of robot decisions to be cautious in times of high environmental perturbation. To demonstrate the ability of the filter, we use it together with an online non-linear controller to perform risk-sensitive output-feedback MPC on various simulated robots, such as a quadrotor subjected to arbitrary changes in its mass, and a KuKa robot facing unforeseen environmental disturbances. Finally, we test the filter on a real quadruped robot Solo12 [19] to perform an external force estimation and balancing task. These experiments demonstrate that the robots are more robust to perturbations with the RS-EKF algorithm than a classical EKF. To the best of our knowledge, this is the first time that a non-linear risk-sensitive output-feedback MPC controller has been deployed on a robot.

II. BACKGROUND

A. Dynamic game output feedback MPC

To design a controller sensitive to the risk related to estimation uncertainty, Whittle [18] introduced a zero-sum game that aims at solving jointly the estimation and control problem. Given a history of measurements $y_{1:t}$, a history of

A. Jordana, A. Meduri and L. Righetti are with New York University, New York. (e-mail: aj2988@nyu.edu, am9789@nyu.edu, lr114@nyu.edu).
J. Carpentier and E. Arlaud are with Inria, Département d'informatique de l'ENS, École normale supérieure, CNRS, PSL Research University, Paris, France (e-mail: justin.carpentier@inria.fr).

control inputs, $u_{0:t-1}$ and a prior on the initial state $\hat{x}_0$, we aim to find a control sequence $u_{t:T-1}$ that minimizes a given cost ℓ while an opposing player aims to find the disturbances $(w_{0:T}, \gamma_{1:t})$ that maximize this cost ℓ minus a weighted norm of the disturbances. Such a problem is formally written as:

$$\min_{u_{t:T-1}} \max_{w_{0:T}} \max_{\gamma_{1:t}} \quad \ell_T(x_T) + \sum_{i=0}^{T-1} \ell_i(x_i, u_i) \tag{1}$$
$$- \frac{1}{2\mu} \left(\omega_0^T P^{-1} \omega_0 + \sum_{j=1}^{t} \gamma_j^T R_j^{-1} \gamma_j + \sum_{j=1}^{T} w_j^T Q_j^{-1} w_j \right)$$

$$\text{subject to} \quad x_0 = \hat{x}_0 + w_0, \tag{2a}$$
$$x_{j+1} = f_j(x_j, u_j) + w_{j+1}, \quad 0 \leq j < T, \tag{2b}$$
$$y_j = h_j(x_j) + \gamma_j, \quad 1 \leq j \leq t. \tag{2c}$$

where $\mu > 0$. x_j is the state, ω_j the process disturbance, y_j the measurement,γ_j the measurement disturbance, T the time horizon, t the current time. The transition model f_j, the measurement model h_j, and the cost ℓ_j are assumed to be $\mathcal{C}^2$. The measurement uncertainty R_j, the process uncertainty Q_j and the initial state uncertainty P are positive-definite matrices. This weighted sum of the disturbances can be seen as the estimation of maximum a posteriori probability (MAP) under Gaussian assumption. Hence, R_j, Q_j and P can be thought of as the covariance matrix of Gaussian noise.

Interestingly, this problem encompasses both formulations of control and estimation. If $t = 0$, in the limit where μ tends to zero, we find the generic OCP formulation [20] which directly minimizes the cost function assuming standard deterministic dynamics. And, if $t = T$ and if we consider all costs ℓ_i to be null, then, (1) is equivalent to maximizing the estimation maximum a posteriori probability (MAP). Here, the parameter μ is referred to as the risk-sensitive parameter and regulates how adversarial the problem is.

Whittle [18] provided a solution to this min-max problem for linear dynamics and quadratic costs. This solution is iteratively obtained with two recursions, one on past disturbances and one on future ones. Recently, [17] showed how this solution could be used to implement an efficient Newton's method that iteratively searches for a saddle point of the more general Problem (1). Exploiting the sparsity of the problem, the proposed Newton step has a linear complexity in the time horizon (a naive optimization has at least a cubic complexity [15]). The solution can then be interpreted as a risk-sensitive Kalman smoother non-trivially coupled to minimax differential dynamic programming (DDP) [21].

Here, we aim to use these insights to derive a computationally efficient risk-sensitive extended Kalman filter that can be used for output feedback MPC. This is done by simplifying Problem (1) to match assumptions common to EKF and DDP while keeping the adversarial min-max formulation.

B. Extended Kalman Filter

The EKF is usually derived by computing the probability a posteriori of the state given measurements, using the linearized dynamics and a Gaussian noise assumption [22]. However, from an optimization point of view, the EKF also corresponds to a Gauss-Newton step around a well-chosen point on the log-likelihood of the MAP [23], i.e. $\log(p(x_t, x_{t-1}|y_t))$. Assuming Gaussian disturbances, $\gamma_t \sim \mathcal{N}(0, R_t)$, $\omega_t \sim \mathcal{N}(0, Q_t)$, the MAP is written as:

$$\max_{x_t, x_{t-1}} \quad -(y_t - h_t(x_t))^T R_t^{-1} (y_t - h_t(x_t))$$
$$-(x_t - f_{t-1}(x_{t-1}, u_{t-1}))^T Q_t^{-1} (x_t - f_{t-1}(x_{t-1}, u_{t-1}))$$
$$-(x_{t-1} - \hat{x}_{t-1})^T P_{t-1}^{-1} (x_{t-1} - \hat{x}_{t-1}) \tag{3}$$

where $\hat{x}_{t-1}$ is the prior knowledge on the past state and P_{t-1} its associated covariance matrix. As shown in [23], a Gauss-Newton step around $\hat{x}_{t-1}$ and $\bar{x}_t = f_{t-1}(\hat{x}_{t-1}, u_{t-1})$ on (3) leads to the well-known recursion [22]:

$$\bar{P}_t = Q_t + F_{t-1} P_{t-1} F_{t-1}^T \tag{4}$$
$$K_t = \bar{P}_t H_t^T (R_t + H_t \bar{P}_t H_t^T)^{-1} \tag{5}$$
$$P_t = (I - K_t H_t) \bar{P}_t \tag{6}$$
$$\Delta\hat{x}_t = K_t (y_t - h_t(\bar{x}_t)) \tag{7}$$
$$\hat{x}_t = \bar{x}_t + \Delta\hat{x}_t \tag{8}$$

where $F_{t-1} = \partial_x f_{t-1}(\hat{x}_{t-1}, u_{t-1})$, $H_t = \partial_x h_t(\bar{x}_t)$, $\hat{x}_t$ is the most likely estimate and P_t the covariance uncertainty.

Notice the similar structure of the costs of Problem (1) and Eq. (3), except that the EKF only uses one measurement and does not include the control cost ℓ_j. Eq. (1) can be seen as a maximization of the estimation log-likelihood up to some cost terms. We leverage this similarity to derive our risk-sensitive EKF. More precisely, we will add cost-dependent terms in the maximization (3) to allow the filter to adapt its estimation to the control objective.

C. Nonlinear MPC

The estimated state can then be used in the MPC module. At each time step, a stagewise cost defined over a horizon H is minimized over future control inputs:

$$\mathcal{L}_t(u_t, \ldots, u_{H-1}) = \ell_{t+H}(x_{T+H}) + \sum_{j=t}^{t+H-1} \ell_j(x_j, u_j) \tag{9}$$

where the state sequence is implicitly defined by the dynamics $x_{j+1} = f_j(x_j, u_j)$ and where x_t is the state estimated by the filter. At each time step, we only use the first control u_t. At the next time step, the state estimate is updated given a new measurement and the OCP is solved again. There are various ways to solve efficiently this problem. A popular algorithm is DDP [24] which reassembles the Newton method but with linear complexity in the time horizon. Additionally, DDP provides a quadratic approximation of the value function which we exploit in our derivation of the risk-sensitive filter.

III. RISK SENSITIVE FILTER

We now introduce RS-EKF, which builds on the dynamic game defined in Eq. (1). First, we modify the game to account for the typical assumptions made for MPC while keeping the adversarial part to ensure a risk-sensitive behavior. Then, we show how to compute the solution with a Gauss-Newton step similar to the EKF, leading to an

algorithm of the same complexity. This results in a modified update step of which the standard EKF is a limit case.

First, as for the EKF, we consider a history of measurements of length one. Then, we disregard future uncertainties and assume deterministic dynamic equations for the future as is done in classical MPC formulations. Indeed, we expect that the high-frequency re-planning will compensate for model discrepancies. Hence, we seek to be adversarial only with respect to the estimation uncertainties. This is written as:

$$\min_{u_{t:t+H-1}} \max_{w_t} \max_{w_{t-1}} \max_{\gamma_t} \mathcal{L}_t(u_t, \dots, u_{H-1}) \tag{10}$$
$$- \frac{1}{2\mu} \left(\gamma_t^T R_t^{-1} \gamma_t + w_t^T Q_t^{-1} w_t + w_{t-1}^T P_{t-1}^{-1} w_{t-1}\right)$$

$$\text{s.t.} \quad x_{t-1} = \hat{x}_{t-1} + w_{t-1}, \tag{11a}$$
$$x_t = f_{t-1}(x_{t-1}, u_{t-1}) + w_t, \tag{11b}$$
$$y_t = h_t(x_t) + \gamma_t. \tag{11c}$$
$$x_{j+1} = f_j(x_j, u_j), \qquad t < j < T. \tag{11d}$$

As presented in [17], one of the key features of the dynamic game is that some of the constraints can be removed with an appropriate change of variable. Indeed, we can use the equality constraints, Eqs. (11a), (11b) and (11c), to replace the disturbance maximization into a maximization over x_{t-1}, x_t:

$$\min_{u_{t:t+H-1}} \max_{x_{t-1}, x_t} \mathcal{L}_t(u_t, \dots, u_{H-1}) \tag{12}$$
$$- \frac{1}{2\mu}(y_t - h_t(x_t))^T R_t^{-1}(y_t - h_t(x_t))$$
$$- \frac{1}{2\mu}(x_t - f_{t-1}(x_{t-1}, u_{t-1}))^T Q_t^{-1}(x_t - f_{t-1}(x_{t-1}, u_{t-1}))$$
$$- \frac{1}{2\mu}(x_{t-1} - \hat{x}_{t-1})^T P_{t-1}^{-1}(x_{t-1} - \hat{x}_{t-1})$$
$$\text{subject to} \quad x_{j+1} = f_j(x_j, u_j), \quad t < j < T,$$

By definition of the MAP [22], this can be written:

$$\min_{u_{t:t+H-1}} \max_{x_{t-1}, x_t} \mathcal{L}_t(u_t, \dots, u_{H-1}) - \frac{1}{\mu} \log(p(x_t, x_{t-1} | y_t)) \tag{13}$$
$$\text{subject to} \quad x_{j+1} = f_j(x_j, u_j), \quad t < j < T.$$

Problem (13) is intractable in the general case. However, by taking the concave-convex assumption, the minimization and maximization can be interchanged according to the minimax theorem. Consequently, the problem is equivalent to:

$$\max_{x_{t-1}, x_t} \quad \log(p(x_{t-1}, x_t | y_t)) + \mu V_t(x_t), \tag{14}$$

where V_t is the value function of the OCP:

$$V_t(x_t) = \min_{u_{t:t+H-1}} \mathcal{L}_t(u_t, \dots, u_{H-1}) \tag{15}$$

Note that in the simplification from Eq. (1) to Eq. (10), it is not necessary to disregard future uncertainties as the value function could be the one resulting from minimax DDP [21]. If $\mu = 0$, we will obtain the unbiased estimate of Kalman filtering and the estimate will be independent of the control objective. Otherwise, if $\mu > 0$, the term $\mu V(x_t)$ will bias the estimate towards regions with higher value function, which in turn will force the controller to be more conservative.

We now take a Gauss-Newton step on the objective of Eq. (14) around the prior: $\hat{x}_{t-1}$ and $\bar{x}_t = f_{t-1}(\hat{x}_{t-1}, u_{t-1})$. $V_t(x_t)$ is independent of x_{t-1} therefore, as shown in the Appendix, the maximization over x_{t-1} can be simplified to:

$$\max_{x_t} -\frac{1}{2}(x_t - \hat{x}_t)^T P_t^{-1}(x_t - \hat{x}_t) \tag{16}$$
$$+ \mu \frac{1}{2}(x_t - \bar{x}_t)^T V_t^{xx}(x_t - \bar{x}_t) + \mu(x_t - \bar{x}_t)^T v_t^x$$

where $\hat{x}_t$ and P_t are defined as in Eq. (8) and (6). where V_t^{xx} (respectively v_t^x) is the hessian (respectively the gradient) of the value function. Those are typically provided by optimal control algorithms such as DDP. In the end, the solution on the maximization over x_t is:

$$\hat{x}_t^{RS} = \bar{x}_t + (I - \mu P_t V_t^{xx})^{-1}(\Delta \hat{x}_t + \mu P_t v_t^x) \tag{17}$$

Interestingly, if $\mu = 0$, we recover the EKF. This was to be expected as, when μ tends to zero, the solution of Problem (1) is exactly the solution of the neutral case where estimation and control are solved independently [18]. Otherwise, the estimate is shifted towards regions with higher cost values. Importantly, the magnitude of the shift depends on P_t the covariance matrix of the estimation. Note that μ cannot be arbitrarily large as $(I - \mu P_{t+1} V_{t+1}^{xx})$ needs to be positive definite. Larger values of μ would make the min-max problem defined in Eq. (1) ill-posed. More details on this limit value can be found in [18]. In the end, the estimate is shifted towards $P_t v_t^x$, i.e. towards a region with a larger cost function, and the magnitude of this shift is increased in the direction corresponding to large eigenvalues of $P_t V_t^{xx}$.

We obtained the solution to the maximization problem (13). Therefore, the cost function can now be minimized with respect to the control inputs by taking $\hat{x}_t^{RS}$ as an initial condition of the OCP, which can be solved with DDP.

Algorithm 1 summarizes the estimation procedure. It can then be used to do output-feedback MPC efficiently. At each time step, given a measurement, past control input, and a quadratic approximation of the value function, a risk-sensitive estimate can be computed. This estimate is then used to minimize the cost function (9) for MPC and the first control input is applied to the real system. Lastly, the quadratic approximation of the value function at $t+1$ is saved as it will be used at the next estimation step.

Algorithm 1: Risk Sensitive EKF

Input: $\hat{x}_{t-1}, u_{t-1}, y_t, P_{t-1}, Q_t, R_t, V_t^x, v_t^x$

/* Predict */

$\bar{P}_t \leftarrow Q_t + F_{t-1} P_{t-1} F_{t-1}^T$

$\bar{x}_t \leftarrow f(\hat{x}_{t-1}, u_{t-1})$

/* Classical Update */

$K_t \leftarrow \bar{P}_t H_t^T (R_t + H_t \bar{P}_t H_T^T)^{-1}$

$P_t \leftarrow (I - K_t H_t) \bar{P}_t$

$\Delta \hat{x}_t \leftarrow K_t(y_t - h_t(\bar{x}_t))$

/* Value function bias */

$p_{x_t} \leftarrow (I - \mu P_t V_t^x)^{-1}(\Delta \hat{x}_t + \mu P_t v_t^x)$

$\hat{x}_t^{RS} \leftarrow \bar{x}_t + p_{x_t}$

Output: $\hat{x}_t^{RS}, P_t$

IV. EXPERIMENTS

We study three problems where we deploy the RS-EKF inside a MPC loop: a planar quadrotor with a load estimation task, a push-recovery experiment on a 7-dof industrial manipulator and lastly, an external force estimation task on a real quadruped robot. In all experiments, we provide a comparison to the nominal case, i.e. the standard EKF with a nominal MPC. To the best of our knowledge, the proposed filter is the first attempt to deploy a robust output-feedback controller on a robot. In each experiment, the OCP is solved with DDP [25]. All the code is available online[1].

A. Planar quadrotor carrying an unknown load

First, we consider a planar quadrotor moving from position $(p_x, p_y) = (0, 0)$ to position $(1, 0)$ while carrying a unknown load during the first half of the itinerary. The robot mass is 2 kg and the mass's load which is unknown a priori is 3 kg. The system dynamics is:

$$\begin{aligned} m\ddot{p}_x &= -(u_1 + u_2)\sin(\theta), \\ m\ddot{p}_y &= (u_1 + u_2)\cos(\theta) - mg, \\ md\ddot{\theta} &= r(u_1 - u_2), \end{aligned} \tag{18}$$

where m is the mass of the robot, d the distance between the rotors, θ the orientation of the quadrotor. u_1 and u_2 are the control inputs representing the force applied at each rotor.

In this experiment, we want to estimate online the mass parameter that changes in the middle of the flying phase. As it is standard in parameter identification [26], we augment the system's state with the unknown parameter and let it be estimated recursively by the filter (EKF or RS-EKF). The state of the system is thus: $x = \begin{pmatrix} p_x & p_y & \theta & \dot{p}_x & \dot{p}_y & \dot{\theta} & m \end{pmatrix}^T$ and it is assumed that $\dot{m} = 0$ up to some random Gaussian noise. The dynamics are integrated with an Euler scheme and a time step of 0.05. We set $P_0 = 10^{-4} I_7$, $R = 10^{-4} I_3$. Q is a 7×7 diagonal matrix where all terms are equal to 10^{-4} except the last one that we set to 2 to represent the uncertainty in the changes of the load. Lastly, we set $\mu = 4 \times 10^{-3}$. The stagewise cost describing the task is:

$$\begin{aligned} \ell(x, u) = \alpha_1 \left(\|p_x - p_x^{des}\|^2 + \|p_y - p_y^{des}\|^2 \right) + \alpha_2 \|\theta\|^2 \\ + \alpha_3 \left(\|\dot{p}_x\|^2 + \|\dot{p}_y\|^2 + \|\dot{\theta}\|^2 \right) + \alpha_4 \|u - \bar{u}\|^2 \end{aligned} \tag{19}$$

where $\bar{u} = \left(\frac{mg}{2}, \frac{mg}{2}\right)^T$ and where: $\alpha_1 = 100$, $\alpha_2 = 10$, $\alpha_3 = 0.01$ and $\alpha_4 = 0.1$. We consider a horizon of 20 nodes and re-plan at each new measurement, i.e. every $0.05s$. Furthermore, we only measure: $y = \begin{pmatrix} p_x & p_y & \theta \end{pmatrix}^T$ to illustrate the estimator capabilities. We simulate $4s$ with both output-feedback MPC controllers: one relying on the standard EKF and the other relying on the RS-EKF.

Figure 1 shows the real mass variation and the estimates of both methods and Figure 2 show the state space trajectories of the quadrotor. It can be seen that the RS-EKF is more reactive when the load is added or dropped. The increase of uncertainty on the components of the state that are important in the cost function leads to mass estimate spikes in the RS-EKF. This overestimation of the mass change in turns leads to an improved control performance (Figure 2). In other words, some of the eigenvalue of $P_t V_t^{xx}$ become larger in the phases of uncertainty, which augments the shift on the estimate as shown in Equation (17). The average Mean Square Error (MSE) relative to the reference trajectory is 0.0011 for the RS-EKF and 0.0024 for the EKF. Hence, risk sensitivity in the estimator brings a 54% improvement in tracking performance. Furthermore, the average cost along the trajectory is 0.0569 for the RS-EKF-based controller and 0.0880 for the EKF-based controller, yielding a 35% improvement. This illustrates how a filter informed of the cost objective can improve the controller's performance.

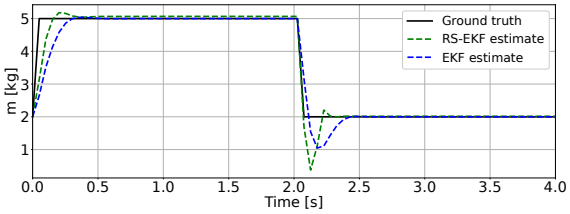

Fig. 1. Mass estimation for both EKF and RS-EKF.

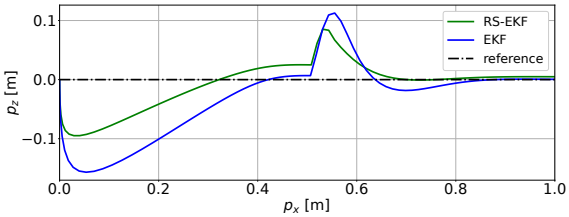

Fig. 2. Quadrotor trajectory for both the EKF-MPC and the RS-EKF-MPC.

B. Kuka robot subject to large external disturbances

Now, we consider the 7-DoF torque-controlled KUKA LWR iiwa R820 14 which needs to track an end-effector reference trajectory. We consider the following task cost:

$$\begin{aligned} \ell_k(x_k, u_k) = 10^{-2}\|x_k - \bar{x}\|_2^2 + 10^{-4}\|u_k - \bar{u}(x_k)\|_2^2 \\ + 10^2 \|p_k^{\text{target}} - \bar{p}(x_k)\|_2^2 \\ \ell_T(x_T) = 10^2 \|p_T^{\text{target}} - \bar{p}(x_k)\|_2^2 + 10^{-2}\|x_k - \bar{x}\|_2^2, \end{aligned} \tag{20}$$

$\bar{x}$, the initial state, is used for regularization and is the concatenation of the initial robot configuration and a 7-dimensional zero vector for the velocity. $\bar{u}(x_k)$ is the gravity term given by the rigid body dynamics. $\bar{p}(x_k)$ is the end-effector position obtained through forward kinematics. p_k^{target} defines an end effector circle trajectory in the xy plane. We use a horizon of 20 collocation point with an integration step of $0.05s$ and re-plan at 500 Hz. We use Pinocchio [27] to compute the robot dynamics and its analytical derivatives.

This experiment aims to showcase the ability of the risk-sensitive filter in bringing conservatism during phases with large environmental perturbations (large forces applied on the end effector). We assume all states are observed with high accuracy, therefore, we set $R = P_0 = 10^{-6} I_{14}$. However, to model the disturbances in the dynamics, we set $Q = 10^{-1} I_{14}$. Finally, we consider $\mu = 7.5 \times 10^4$.

[1] https://github.com/machines-in-motion/risk-sensitive-EKF

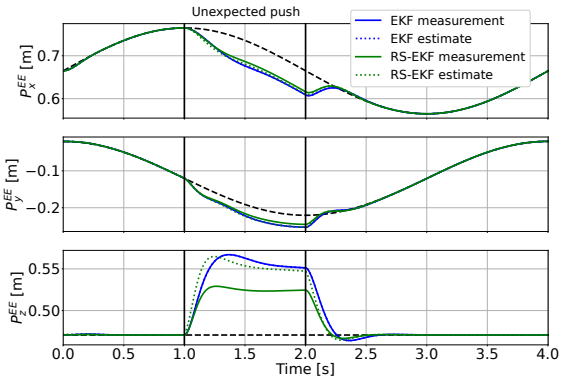

Fig. 3. End effector trajectory on a tracking task for both the EKF-MPC and the RS-EKF-MPC. An unexpected force is applied between $1s$ and $2s$.

Figure 3 depicts the end effector trajectory for both controllers and their respective estimates. An external force of norm $80N$ is applied on the end-effector in the x and z direction from time $1s$ to $2s$. The RS-EKF overestimates the distance between the reference and the end-effector which leads to a more aggressive response of the controller and results in the end-effector remaining closer to the reference. This illustrates how taking a pessimistic estimate with respect to the cost can improve control performance. Note that both estimates are state estimates that we projected in the end effector space, the space where the cost function is defined, to draw Fig. 3. This illustrates the ability of the method to handle nonlinear dynamics and cost functions.

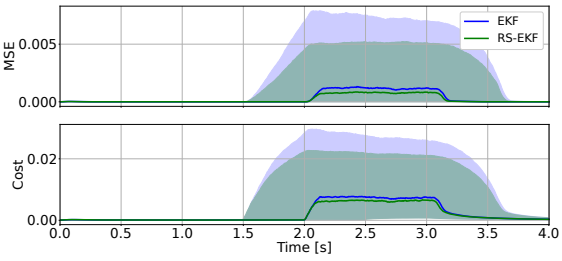

Fig. 4. Median MSE on the tracking over $10,000$ experiments with random external disturbances. The envelope represents the 25th and 75th percentiles.

To validate the consistency of the filter, we performed $10,000$ experiments where the timing and direction of the forces are uniformly sampled with a fixed perturbation duration of $1s$ and force magnitude of $80N$. Fig. 4 shows the median end effector error trajectory. In average, RS-EKF brings a 32% improvement in the MSE and a 22% reduction of the mean cost.

C. Load estimation on a quadruped robot

Finally, we deploy the RS-EKF on a real 12-DOFs, torque-controlled quadruped robot - Solo12 [19]. We demonstrate the superior performance of the RS-EKF in estimating external wrenches while the robot is standing. A non-linear MPC scheme is used to generate the standing behavior. At each control cycle, we minimize a cost function using a centroidal model to compute the optimal forces and trajectory that keep the robot's base at a desired height and orientation.

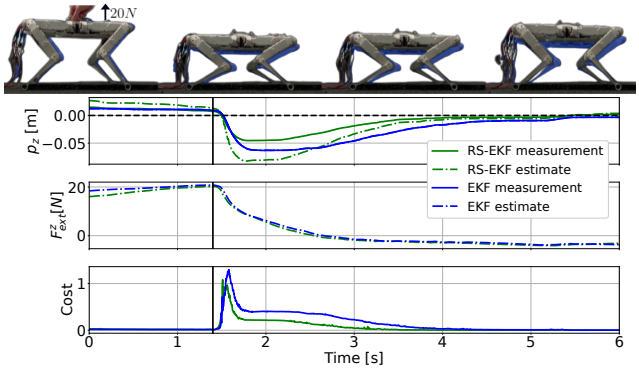

Fig. 5. Comparison of both methods after an external force of 20 N is applied by pulling the robot vertically. Top figure: overlay of the robot movements using both the EKF (dark blue) and the RS-EKF (solid). Bottom figure: RS-EKF and EKF estimates and real measurements. The vertical line indicates the moment when the robot is dropped.

Additionally, we use an augmented state to estimate the external wrench applied to the robot as in [28]:

$$\dot{c} = \frac{1}{m} l, \quad \dot{F}_{\text{ext}} = 0, \quad \dot{\tau}_{\text{ext}} = 0, \tag{21}$$
$$\dot{l} = mg + \sum_{i=1}^{M_c} F_i + F_{\text{ext}} \quad \dot{k} = \sum_{i=1}^{M_c} (p_i - c) \times F_i + \tau_{\text{ext}}$$

where m denotes the mass, M_c the number of end effectors in contact andp_i are contact locations. The state is $x = \begin{pmatrix} c & l & k & F_{\text{ext}} & \tau_{\text{ext}} \end{pmatrix}^T$ which includes the center of mass (c), linear momentum (l), angular momentum (k) and external wrench ($F_{\text{ext}}, \tau_{\text{ext}}$). The measurement is $y = \begin{pmatrix} c & l & k \end{pmatrix}^T$ up to some noise. Motion capture measures the base position, velocity, and orientation and an IMU gives the orientation velocity. Joint encodings are provided and their velocities are derived with finite differences. Given $q, \dot{q}$, we compute c, l, k which are used as the measurement by the filter.

The control input, $u = \begin{pmatrix} F_1 & \dots & F_{M_c} \end{pmatrix}^T$, is a $M_c \times 3$ dimensional vector, representing the force applied at each end effector. For this experiment, the robot is standing, therefore, $M_c = 4$. The cost function for the OCP is:

$$\ell_t(x, u) = \|x - x^\star\|_{H_x} + \|u - u^\star\|_{H_u} + 10^5 \sum_{i=1}^{M_c} \ell_{\text{barrier}}(u_{3i})$$
$$\ell_T(x) = (x - x^\star)^T H_x (x - x^\star) \tag{22}$$

where $H_x = \text{BlockDiag}(10^2 I_3, 10 I_6)$ and H_u is a diagonal matrix where the diagonal terms are made of M_c times the following sequence $(10^{-4}, 10^{-4}, 10^{-6})$. Lastly, ℓ_{barrier} is a quadratic barrier function that creates a soft constraint on the maximum forces the robot can apply on the ground. More specifically, $\ell_{\text{barrier}}(u)$ is equal to u^2 if $u < 0$, $(u - 10)^2$ if $u > 10$ or 0 otherwise. Here, $x^\star$ is designed to keep the CoM at constant height above the ground and the base horizontal, $u^\star$ is gravity compensation. The reference desired angular momentum for the OCP is adapted to bring the base back to a horizontal position as in [29]. We set $k^* = \frac{1}{T} log_3(R_t R_{\text{des}}^T)$, where R_t and R_{des} are the current and desired base rotation matrices respectively and T the horizon length. log_3 is the matrix logarithm mapping an element of $SO(3)$ to $\mathfrak{so}(3)$.

We solve this OCP at 100 Hz using the DDP solver Croccodyl [25] and track the desired forces using a task space inverse dynamics QP [30] that we solve at 1 kHz using [31]:

$$\begin{aligned} \min_{f,\tau,a} \quad & \frac{1}{2}\|f - F\|^2 \\ \text{subject to} \quad & Ma + g = J^T f + S^T \tau + S^T f_{\text{friction}} \\ & Ja = -\dot{J}\dot{q}, \end{aligned} \tag{23}$$

where J is the contact Jacobian, M the mass matrix, S the selection matrix that projects on the actuated joints, and g is the generalized gravity vector at the current time step. Static friction in each joint was estimated to 0.07. However, to keep a continuous model, we write $f_{\text{friction}} = -0.07\frac{2}{\pi}\arctan(2S\dot{q})$. The first constraints ensure dynamics consistency and the second ensures that the end effectors do not move. We update our state estimate using the filters at 200 Hz with $\mu = 6$. For both filter, we consider the following parameters: $P_0 = Q = \text{BlockDiag}(10^{-3}I_6, 10^{-4}I_3, 10^{-1}I_3, 10^{-2}I_3)$ $R = \text{BlockDiag}(10^{-4}I_3, 10^{-2}I_3, 10^{-4}I_3)$.

Fig. 5 shows results from the first experiment where the base of Solo12 is pulled up (in the z direction) until an external estimate of 20N is computed by both filters (vertical line at time $1.4s$). The base is then released to let Solo12 recover and bring its base back to the desired height. The RS-EKF helps the OCP to react quicker and bring the base to the desired location sooner. This happens because the RS-EKF, during periods of high uncertainty, underestimates the base height in z as compared to EKF, which makes the OCP generate higher ground reaction forces to bring the base up sooner. As it can be seen in Fig. 5, the external vertical force does not converge exactly to zero. We find experimentally that this is due to friction. Lastly, the cost of the RS-EKF-based controller is lower after the robot is dropped. The average cost of the RS-EKF is 0.065 while the one of the EKF is 0.130, which corresponds to a 50% improvement.

To get rid of the human error, we perform two additional experiments where the filters are initialized with exactly the same priors. First, we initialize both filters with a wrong prior on the external vertical force of 20 N, while in reality, no force is applied on the robot. This experiment creates an identical situation as the previous experiment while also ensuring the exact same initial conditions. The results are shown in Fig 6, where the RS-EKF still performs better. In that experiment, we obtain a 62.9% improvement in the average cost. In the second experiment, we initialize the

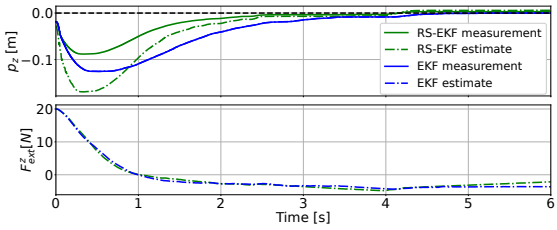

Fig. 6. Comparison of the RS-EKF and EKF when initialized with a wrong prior of 20 N on the estimated vertical external force.

filters with a wrong prior of -10 N on the external force, while, in fact, there is no force on the robot. The RS-EKF reacts sooner than EKF once again. It brings the base of Solo12 back to the desired location sooner than EKF (Fig. 7). We also obtain a 58.9% improvement in the average cost.

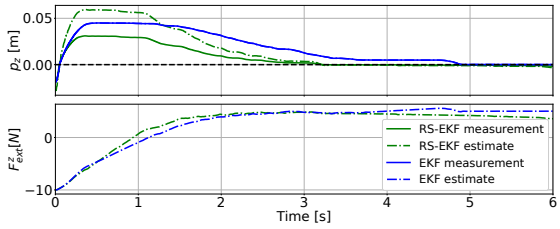

Fig. 7. Comparison of the RS-EKF and EKF when initialized with a wrong prior of -10 N on the estimated vertical external force.

V. CONCLUSION

Leveraging recent results in dynamic game control, we introduced a risk-sensitive variation of the EKF which biases estimates towards high regions of the control cost which result in more robust controllers. Furthermore, the complexity of this filter is similar to the EKF. Experiments both in simulation and a real robot show the benefits of this filter for output-feedback MPC in face of high uncertainty.

APPENDIX

By taking a quadratic approximation of the value function, the Gauss-Newton step can be written as:

$$\begin{aligned} \max_{x_{t-1}} \max_{x_t} \quad & \mu(x_t - \bar{x}_t)^T V_t^{xx}(x_t - \bar{x}_t) + 2\mu(x_t - \bar{x}_t)^T v_t^x \\ & - (\Delta y - H_t \Delta x_t)^T R_t^{-1}(\Delta y - H_t \Delta x_t) - \Delta x_{t-1}^T P_{t-1}^{-1} \Delta x_{t-1} \\ & - (\Delta x_t - F_{t-1}\Delta x_{t-1})^T Q_t^{-1}(\Delta x_t - F_{t-1}\Delta x_{t-1})) \end{aligned} \tag{24}$$

where $\Delta y = y_t - h(\hat{x}_t)$, $\Delta x_{t-1} = x_{t-1} - \hat{x}_{t-1}$, $\Delta x_t = x_t - \hat{x}_t$. It can then be found that $x_{t-1} = \tilde{Q}^{-1}\tilde{q}$, where:

$$\begin{aligned} \tilde{Q} &= P_{t-1}^{-1} + F_{t-1}^T Q_t^{-1} F_{t-1} \\ \tilde{q} &= -P_{t-1}^{-1}\hat{x}_{t-1} - F_{t-1}^T Q_t^{-1}(x_t - \hat{x}_t) - F_{t-1}^T Q_t^{-1} F_{t-1}\hat{x}_{t-1} \end{aligned} \tag{25}$$

by using the Woodbury lemma [22], It can be shown that:

$$\begin{aligned} \max_{x_t} \quad & \mu\frac{1}{2}(x_t - \bar{x}_t)^T V_t^{xx}(x_t - \bar{x}_t) + \mu(x_t - \bar{x}_t)^T v_t^x \\ & - \frac{1}{2}(\Delta y - H_t\Delta x_t)^T R_t^{-1}(\Delta y - H_t \Delta x_t) - \frac{1}{2}\Delta x_t^T \bar{P}_t^{-1}\Delta x_t \end{aligned}$$

where $\bar{P}_t$ is defined as in (4). Finally, we can show that:

$$\begin{aligned} \max_{x_t} \quad & \mu\frac{1}{2}(x_t - \bar{x}_t)^T V_t^{xx}(x_t - \bar{x}_t) + \mu(x_t - \bar{x}_t)^T v_t^x \\ & - \frac{1}{2}(x_t - \hat{x}_t - \Delta\hat{x}_t)^T P_t^{-1}(x_t - \hat{x}_t - \Delta\hat{x}_t) \end{aligned} \tag{26}$$

where $P_t, \Delta\hat{x}_t$ are defined as in (6), (7).

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation grants 1932187, 2026479, 2222815 and 2315396, and by the French government under the management of Agence Nationale de la Recherche through the NIMBLE project (ANR-22-CE33-0008) and as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

REFERENCES

[1] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014.

[2] S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous robots*, vol. 40, pp. 429–455, 2016.

[3] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[4] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[5] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[6] E. Dantec, M. Naveau, P. Fernbach, N. Villa, G. Saurel, O. Stasse, M. Taïx, and N. Mansard, "Whole-body model predictive control for biped locomotion on a torque-controlled humanoid robot," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 638–644.

[7] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[8] T. Erez, K. Lowrey, Y. Tassa, V. Kumar, S. Kolev, and E. Todorov, "An integrated system for real-time model predictive control of humanoid robots," in *2013 13th IEEE-RAS International conference on humanoid robots (Humanoids)*. IEEE, 2013, pp. 292–299.

[9] J. Koenemann, A. Del Prete, Y. Tassa, E. Todorov, O. Stasse, M. Bennewitz, and N. Mansard, "Whole-body model-predictive control applied to the hrp-2 humanoid," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 3346–3351.

[10] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 1–9.

[11] S. Kleff, A. Meduri, R. Budhiraja, N. Mansard, and L. Righetti, "High-frequency nonlinear model predictive control of a manipulator," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7330–7336.

[12] G. A. Einicke and L. B. White, "Robust extended kalman filtering," *IEEE Transactions on Signal Processing*, vol. 47, no. 9, pp. 2596–2599, 1999.

[13] N. A. Villa and P.-B. Wieber, "Model predictive control of biped walking with bounded uncertainties," in *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 836–841.

[14] A. Gazar, M. Khadiv, A. Del Prete, and L. Righetti, "Stochastic and robust mpc for bipedal locomotion: A comparative study on robustness and performance," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 61–68.

[15] D. A. Copp and J. P. Hespanha, "Simultaneous nonlinear model predictive control and state estimation," *Automatica*, vol. 77, pp. 143–154, 2017.

[16] J. Köhler, F. Allgöwer, and M. A. Müller, "A simple framework for nonlinear robust output-feedback mpc," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 793–798.

[17] A. Jordana, B. Hammoud, J. Carpentier, and L. Righetti, "Stagewise newton method for dynamic game control with imperfect state observation," *IEEE Control Systems Letters*, vol. 6, pp. 3241–3246, 2022.

[18] P. Whittle, "Risk-sensitive linear/quadratic/gaussian control," *Advances in Applied Probability*, vol. 13, no. 4, pp. 764–777, 1981.

[19] F. Grimminger, A. Meduri, M. Khadiv, J. Viereck, M. Wüthrich, M. Naveau, V. Berenz, S. Heim, F. Widmaier, T. Flayols *et al.*, "An open torque-controlled modular robot architecture for legged locomotion research," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3650–3657, 2020.

[20] M. C. Campi and M. R. James, "Nonlinear discrete-time risk-sensitive optimal control," *International Journal of Robust and Nonlinear Control*, vol. 6, no. 1, pp. 1–19, 1996.

[21] J. Morimoto and C. Atkeson, "Minimax differential dynamic programming: An application to robust biped walking," *Advances in neural information processing systems*, vol. 15, 2002.

[22] S. Thrun, "Probabilistic robotics," *Communications of the ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[23] B. M. Bell and F. W. Cathey, "The iterated kalman filter update as a gauss-newton method," *IEEE Transactions on Automatic Control*, vol. 38, no. 2, pp. 294–297, 1993.

[24] D. Murray and S. Yakowitz, "Differential dynamic programming and newton's method for discrete optimal control problems," *Journal of Optimization Theory and Applications*, vol. 43, no. 3, pp. 395–414, 1984.

[25] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, L. Righetti, S. Vijayakumar, and N. Mansard, "Crocoddyl: An efficient and versatile framework for multi-contact optimal control," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2536–2542.

[26] H. U. Voss, J. Timmer, and J. Kurths, "Nonlinear dynamical system identification from uncertain and indirect measurements," *International Journal of Bifurcation and Chaos*, vol. 14, no. 06, pp. 1905–1933, 2004.

[27] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library: A fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *2019 IEEE/SICE International Symposium on System Integration (SII)*. IEEE, 2019, pp. 614–619.

[28] N. Rotella, A. Herzog, S. Schaal, and L. Righetti, "Humanoid momentum estimation using sensed contact wrenches," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 556–563.

[29] A. Meduri, P. Shah, J. Viereck, M. Khadiv, I. Havoutis, and L. Righetti, "Biconmp: A nonlinear model predictive control framework for whole body motion planning," *IEEE Transactions on Robotics*, 2023.

[30] A. Herzog, N. Rotella, S. Mason, F. Grimminger, S. Schaal, and L. Righetti, "Momentum control with hierarchical inverse dynamics on a torque-controlled humanoid," *Autonomous Robots*, vol. 40, pp. 473–491, 2016.

[31] A. Bambade, S. El-Kazdadi, A. Taylor, and J. Carpentier, "Prox-qp: Yet another quadratic programming solver for robotics and beyond," in *RSS 2022-Robotics: Science and Systems*, 2022.