ARMCMC: ONLINE MODEL PARAMETERS DENSITY ESTIMATION IN BAYESIAN PARADIGM

Anonymous authors
Paper under double-blind review

ABSTRACT

Although the Bayesian paradigm provides a rigorous framework to estimate the full probability distribution over unknown parameters, its online implementation can be challenging due to heavy computational cost. This paper proposes Adaptive Recursive Markov Chain Monte Carlo (ARMCMC) which estimates full probability density of model parameters while alleviating shortcomings of conventional online approaches. These shortcomings include only being able to account for Gaussian noise, only being applicable to systems with linear in the parameters (LIP) constraint, and having requirements on persistence excitation (PE). In ARMCMC, we propose a variable jump distribution, which depends on a temporal forgetting factor. This allows one to adjust the trade off between exploitation and exploration, depending on whether there is an abrupt change to the parameter being estimated. We prove that ARMCMC requires fewer samples to achieve the same precision and reliability compared to conventional MCMC approaches. We demonstrate our approach on a challenging benchmark: estimation of parameters in the Hunt-Crossley dynamic model, which models both on/off contact forces applied to soft materials. Our method shows up to 75% improvement in parameter point estimation accuracy and approximately 40% reduction in tracking error compared with recursive least squares and conventional MCMC.

1 INTRODUCTION

Bayesian methods are powerful tools to not only obtain a numerical estimate of a parameter but also give a measure of confidence (Kuśmierczyk et al., 2019; Bishop, 2006; Joho et al., 2013). In particular, Bayesian inference calculates the probability distribution of parameters rather than a point estimate which is prevalent in frequentist paradigms (Tobar, 2018). One of the main advantages of probabilitic frameworks is that they enable decision making under uncertainty (Noormohammadi-Asl & Taghirad, 2019). In addition, knowledge fusion is significantly facilitated in probabilitic frameworks; different sources of data or observations can be combined according to their level of certainty in a principled manner (Agand & Shoorehdeli, 2019). Nonetheless, Bayesian inference requires high computational effort for obtaining the whole probability distribution, and a priori general knowledge requirement on noise distribution before estimation.

One of the most effective methods for Bayesian inferrence is Markov chain Monte Carlo methods (MCMC). In the field of system identification, MCMC variants such as the one recently proposed by Green (2015) mostly focused on offline system identification. This is partly due to computational challenges which prevent real-time use (Kuindersma et al., 2012). On the other hand, the main concerns in online problems were mostly due to the distribution or expectation maximization instead of full probability evaluation (Ninness & Henriksen, 2010).

Motion filtering and force prediction of robotic manipulators are important fields of study with interesting challenges suitable for Bayesian inference to address (Saar et al., 2018). Here, measurements are inherently noisy, which is not desirable for control purposes. Besides, inaccuracy, inaccessibility, and cost are the typical shortcomings that make them not ideal for solely use (Agand et al., 2016). Different environment identification methods have been proposed in the literature for linear and Gaussian noise (Wang et al., 2018); however, in the case of a nonlinear model like Hunt-Crossley with non-Gaussian noise (e.g. impulsive disorder), there is no optimal solution for the identification problem. Diolaiti et al. (2005) proposed a double-stage bootstrapped method for online identification

of the Hunt-Crossley model, which is sensitive to parameter initial conditions. In addition, convergence can be slow due to the propagation of error from one stage to the next. A single-stage method for estimation of the Hunt-Crossley model is proposed by Haddadi & Hashtrudi-Zaad (2012); in this case, some restrictive conditions must be met to calculate the parameters. Moreover, the method does not offer any solution when there are discontinuous changes in the dynamic model, which is common in the transition phase from contact to free motion and vice versa. (Williams et al., 2018).

This paper proposes a new method, Adaptive Recursive Markov Chain Monte Carlo (ARMCMC), to address some of the weaknesses of traditional online identification methods such as only being appllicable to systems Linear in Parameter (LIP), having Persistent Excitation (PE) requirements, and assuming Gaussian noise. ARMCMC is an online method that takes advantage of the previous posterior distribution, as long as there is no sudden change in the parameter distribution. To achieve this, we define a new *variable jump distribution* that accounts for the degree of model mismatch using a *temporal forgetting factor*. The temporal forgetting factor is computed from a model mismatch index, and determines whether ARMCMC employs modification or reinforcement to either restart or refine parameter distribution. We demonstrate our method on the Hunt-Crossley model, a well-known model of soft-contact environments, and show favorable perforrmance compared to state-of-the-art baselines.

The rest of this paper is organized as follows. In Sec. 2, some introductory context about Bayesian, MCMC, and Hunt-Crossley model is presented. Sec 3 is devoted to presenting the proposed ARMCMC approach in a step-by-step algorithm. Empirical results on a reality-based model of a soft contact environment capturing Hunt-Crossley dynamic are presented in Sec. 4. Finally, the concluding remarks and future directions are granted in Sec 5.

2 PRELIMINARIES

2.1 PROBLEM STATEMENT

In the Bayesian paradigm, estimates of parameters are given in the form of the posterior probability density function (pdf) over the parameters; this pdf can be continuously updated as new data points are received. Consider the following general model:

$$Y = F(X, \theta) + \nu, \tag{1}$$

where Y, X, and ν are concurrent output, input, and noise vectors, respectively. To calculate the posterior probability, the observed data along with a prior distribution are combined via Bayes' rule (Khatibisepehr et al., 2013). The data includes input/output data pairs $(X, Y)_t$. For convenience, we define $D^t = \{(X, Y)_{t_m+1}, (X, Y)_{t_m+2}, \cdots, (X, Y)_{t_m+N_s+1}\}$. We will be applying updates to the posterior pdf using batches of data points; hence, it will be convenient to partition data as follows:

$$D_{t_0}^{t_f} = [D^{t-1}, D^t] = [D_{t_0}^{t_m}, D_{t_{m+1}}^{t_{m+N_s+1}}], \tag{2}$$

where $N_s = T_s/T = t_m - t_0$ is the number of data points in each data pack $D_{t_m+1}^{t_{m+N_s+1}}$ with T, T_s being the data and algorithm sampling times, respectively. This partitioning is convenient for online applications, as $D_{t_0}^{t_m}$ should have been previously collected and from t_{m+1} to t_{m+N_s+1} the algorithm is executed. Finally, computation of the results' inference are completed at the time step t_{m+N_s+2}. According to Bayes' rule and assuming data points are i.i.d in equation 1, we have

$$P(\theta^t|[D_{t_0}^{t_{m+N_s+1}}]) = \frac{P\big(D_{t_{m+1}}^{t_{m+N_s+1}}|\theta^t, D_{t_0}^{t_m}\big)P(\theta^t|D_{t_0}^{t_m})}{\int P\big(D_{t_{m+1}}^{t_{m+N_s+1}}|\theta^t, D_{t_0}^{t_m}\big)P(\theta^t|D_{t_0}^{t_m})d\theta^t}, \tag{3}$$

where θ^t denote the parameters at time step t. $P(\theta^t|D_{t_0}^{t_m})$ is the prior distribution over parameters at time t_m, which is also the posterior distribution at the previous algorithm sampling time. $P\big(D_{t_{m+1}}^{t_{m+N_s+1}}|\theta^t, D_{t_0}^{t_m}\big)$ is the likelihood function which is obtained by the one-step-ahead prediction:

$$\hat{Y}^{t|t-1} = F(D^{t-1}, \theta^t), \tag{4}$$

where $\hat{Y}^{t|t-1}$ is the prediction of the output in (1), θ^t is the current identified model parameters. If the model in (4) is valid, then the difference between the real output and predicted should be

measurement noise, (i.e., $Y^{t|t-1} - \hat{Y}^{t|t-1} = \nu$). Therefore, the model parameter may be updated as follows:

$$P\big(D^t|\theta^t, D^{t-1}\big) = \prod_{t_m+1}^{t_m+N_s+1} P_\nu\big(Y^{t|t-1} - \hat{Y}^{t|t-1}\big), \tag{5}$$

where P_ν is the probability distribution of noise. Note that there is no restriction on the types of the noise probability distribution.

Remark 1: As it was mentioned before, there is no need to know the exact probability distribution of noise. This probability distribution can be simply substituted with a Gaussian distribution given some sense of data noise mean and variance which can be obtained easily with preprocessing (Bishop, 2006).

Remark 2: To avoid numerical issues, the summation of probability logarithms are calculated. In addition, each pair in algorithm time sample is weighted based on its temporal distance to concurrent time. Therefore Eq. (5) is modified as

$$\begin{aligned} \log\big(P(\cdot)\big) &= \sum_{t_m+1}^{t_m+N_s+1} \log P_\nu(e^t), \\ e_n^t &= \Big(Y_n^t - F^t(D^{t-1}(n), \theta^t)\Big) e^{-\rho(N_s-n)}, \end{aligned} \tag{6}$$

where ρ is a design parameter, and $e^t = [e_1^t, ..., e_n^t, ..., e_{N_s}^t]$. We are trying to solve the Bayesian optimization problem, that gives us the pdf of the model parameters (θ) given the data pairs in the presence of uncertainty.

2.2 Markov Chain Monte Carlo

MCMC is often employed to compute the posterior pdf numerically. The multidimensional integral in (3) is approximated by samples drawn from the posterior pdf. The samples are first drawn from a different distribution called proposal distribution, denoted $q(.)$, which can be sampled from much more easily compared to the posterior. Brooks et al. (2011) discuss different types of MCMC implementation which may employ various proposal distributions and corresponding acceptance criteria. The main steps of the Metropolis-Hastings algorithm as a popular approach are listed following (Ninness & Henriksen, 2010):

1. Set initial guess θ_0 while $P(\theta_0|Y) > 0$ for iteration $k = 1$,
2. Draw candidate parameter θ_{cnd}, at iteration k, from the proposal distribution, $q(\theta_{cnd}|\theta_{k-1})$
3. Compute the acceptance probability,
$$\alpha(\theta_{cnd}|\theta_{k-1}) = \min\Big\{1, \frac{P(\theta_{cnd}|D)q(\theta_{k-1}|\theta_{cnd})}{P(\theta_{k-1}|D)q(\theta_{cnd}|\theta_{k-1})}\Big\}, \tag{7}$$
4. Generate a uniform random number γ in $[0, 1]$,
5. 'Accept' candidate if $\gamma \leq \alpha$ and 'ignore' it if $\gamma > \alpha$,
6. Set iteration to $k + 1$ and go to step 2.

2.3 Precision and Reliability

Two important notions in probabilistic framework to compare results are precision (ϵ) and reliability (δ). The former represents the closeness of a sample to the ground truth, and the latter represents the probability that an accepted sample lies within ϵ of the ground truth.

Lemma: Let P_k be the pdf induced by k samples, and $\mathbb{E}(P_k)$ denotes its expected value. According to Chernoff bound (Tempo et al., 2012), given some $\epsilon, \delta \in [0, 1]$, if the number of samples (k) in MCMC satisfies

$$k \geq \frac{1}{2\epsilon^2} \log(\frac{2}{1-\delta}) \Rightarrow \tag{8}$$

$$Pr\Big\{\{P_k - \mathbb{E}(P_k)\} \leq \epsilon\Big\} \geq \delta. \tag{9}$$

2.4 CONTACT DYNAMIC MODEL

Consider the dynamic of contact as described by the Hunt-Crossley model which is more consistent with the physics of contact than classical linear models such as Kelvin-Voigt Haddadi & Hashtrudi-Zaad (2012). In order to overcome the shortcomings of linear models, Hunt & Crossley (1975) proposed the following hybrid/nonlinear model :

$$f_e(x(t)) = \begin{cases} K_e x^p(t) + B_e x^p(t)\dot{x}(t) & x(t) \geq 0 \\ 0 & x(t) < 0 \end{cases}, \tag{10}$$

in which $K_e, B_e x^p$ denote the nonlinear elastic and viscous force coefficients, respectively. The parameter p is typically between 1 and 2, depending on the material and the geometric properties of contact. Also $x(t), \dot{x}(t), f_e$ are the current position, velocity (as inputs X) and contact force (as output (Y in (1)) of a needle near or inside the soft material, with $x \geq 0$ representing the needle being inside. K_e, B_e, n are three unknown parameters (θ in Eq. (1)) that needs to be estimated. An online estimate of environment force plays a pivotal role in stable interaction between robotic manipulators and unknown environments.

To make the parameters ready for the regression problem we have

$$\begin{aligned} \log(f_e) &= \log(K_e x_s^p + B_e \dot{x}_s x_s^p), \\ \log(f_e) &= p\log(x_s) + \log(K_e + B_e \dot{x}_s). \end{aligned} \tag{11}$$

We will use the RLS method proposed in (Haddadi & Hashtrudi-Zaad, 2012) as a baseline of comparison. However, in this case, it is needed to make the assumption that $B_e/K_e\dot{x}_s << 1$. With this assumption, we have

$$\begin{aligned} \log(1 + B_e/K_e\dot{x}_s) &\approx B_e/K_e\dot{x}_s, \\ \log(f_e) &= p\log(x_s) + \log(K_e) + B_e/K_e\dot{x}_s. \end{aligned} \tag{12}$$

Hence, the regressors and the parameters are obtained as follows:

$$\begin{aligned} \phi &= [1, \dot{x}_s, ln(x_s)], \\ \theta &= [\log(K_e), B_e/K_e, p]^T. \end{aligned} \tag{13}$$

Moreover, it should be noticed that the vector of parameters (θ) in Eq. (13) are not independent, which may lead to divergence.

3 ARMCMC ALGORITHM

At every time, ARMCMC recursively estimates the posterior distribution by drawing samples. The number of samples drawn is constrained by the desired precision and reliability, and the real-time requirement. On the other hand, the maximum number of data points in each data pack, N_s, is limited by the frequency of model variation, and the minimum is confined by the minimum time such that the algorithm is considered real time. We propose a variable jump distribution that enables both enriching or exploring. This will necessitate definition of temporal forgetting factor as a measure to reflect current underlying dynamic of the data.In other words, this parameter will show validity of previous model for the current data. We also prove that ARMCMC achieves the same precision and reliability with fewer samples compared to traditional MCMC. Algorithm 1 summarizes ARMCMC.

3.1 VARIABLE JUMP DISTRIBUTION

We propose a variable jump distribution (also known as proposal distribution) to achieve faster convergence, thereby enabling real-time parameter estimation:

$$q_k^t(\theta^t|\theta_{k-1}^t) = \begin{cases} P(\theta^{t-1}|D^{t-1}) & \lambda_k \leq \lambda^t \\ N(\mu_D, \sigma_\nu) & \lambda_k > \lambda^t \end{cases}, \tag{14}$$

where θ_{k-1}^t is the $(k-1)$-th parameter sample given the t-th data pack throughout the MCMC evaluation. Averaging the second half of this quantity will constructs θ^t. $P(\theta^{t-1}|D^{t-1})$ is the posterior distribution of the parameters at the previous algorithm time step, and $N(\mu_D, \sigma_\nu)$ is a Gaussian distribution with μ_D, σ_e computed using sample-based mean and variance of D^{t-1}.

Algorithm 1 ARMCMC

Assumptions: 1) roughly noise mean (μ_ν) 2) roughly noise variance (σ_ν) 3) desired precision and reliability (ϵ_0, δ_0) 4) desired threshold for model mismatch (ζ_{th})
Goal: Online calculation of parameters posterior distribution given the consecutive t-th pack of data ($P(\theta^t|D^t)$)
Initialization: Prior knowledge for θ_1^0, n=0
Consider desire precision and reliability (ϵ, δ)
repeat
  Put $t_0 = n * N_s + 1$ from (2), n++
  Add new data pack to dataset D^t
  Model mismatch index: ζ^t from (16)
  if $\zeta^t < \zeta_{th}$ **then**
    Reinforcement: set prior knowledge equal to the latest posterior of previous pack
    Temporal forgetting factor: λ^t from (15)
  else
    Modification: set prior knowledge θ_1^n
    Temporal forgetting factor: $\lambda^t = 0$
  end if
  Set maximum iteration k_{max} from (17)
  for $k = 1$ **to** k_{max} **do**
    Proposal distribution:
      • draw $\lambda_k \sim U(0, 1)$
      • *Variable jump distribution:* $q_k^t(.)$ from (14)
    Draw $\theta_k^{t*} \sim q_k^t(.)$
    Acceptance rate: $\alpha(.)$ from (7)
    Draw $\gamma \sim U(0, 1)$
    if $\gamma \leq \alpha$ **then**
      'Accept' the proposal
    end if
  end for
  Wait to build $D_{t_0}^{t_{m+N_s+1}}$ (algorithm sample time)
until No data is obtained

The hyperparameter λ^t, the *temporal forgetting factor*, is an adaptive threshold for the t-th pack that takes inspiration from in the classical system identification terminology; it regulates how previous knowledge affects the posterior distribution. Smaller values of λ^t intuitively means there may be a large sudden change in θ, and thus more exploration is needed. Conversely, larger values of λ^t is appropriate when θ is changing slowly, and thus previous knowledge should be exploited. As more data is obtained, better precision and reliability can be achieved.

3.2 Temporal forgetting factor

Depending on whether the distribution of the parameter θ has changed significantly, a new sample is drawn according to the *modification* or the *reinforcement* mode. Reinforcement is employed to make the identified probability distribution more precise when it is not undergoing sudden change. Modification is employed otherwise to re-identify the distribution "from scratch". Therefore, we define a *model mismatch index*, denoted ζ^t, such that when it surpass a predefined threshold ($\zeta^t > \zeta_{\text{th}}$), modification is applied. Otherwise, if $\zeta^t \leq \zeta_{\text{th}}$, then ζ^t is used to determine λ^t as follows:

$$\lambda^t = e^{-|\mu_\nu - \zeta^t|}, \tag{15}$$

where μ_ν is an estimation of the noise mean by calculation expectation value in relation (1). Note that employing modification is equivalent to setting $\lambda^t = 0$. The model mismatch index ζ^t itself is calculated by averaging over the error of the previous model given the current data

$$\zeta^t = 1/N_s \sum_{n=1}^{N_s} \Big(y_n^t - \mathop{\mathbb{E}}_{\theta \in \theta^{t-1}} (F(D^t(n), \theta)) \Big), \zeta^0 = \infty \tag{16}$$

Remark 3: Since parameter distribution estimation is obtained with a delay of two algorithm time steps sample delay (first for obtaining D^{t-1} and second for running the algorithm), a two-stage approaches such as *Expected Maximization* (EM) are suggested to compensate it. (For more information please refer to Ravichandar & Dani (2017)).

3.3 MINIMUM REQUIRED EVALUATION

Theorem 1. *Let ϵ and δ are given as the desire precision and reliability and the initial sample has enough number of evaluation (as in (8)). To satisfy the inequality in Eq. (8), the minimum number of samples k in ARMCMC is calculated using this implicit relation*

$$k_{min} = \frac{1}{2\epsilon^2} \log\left(\frac{2}{\lambda^t(1-\delta) + 2(1-\lambda^t)e^{-2\epsilon^2(1-\lambda^t)k_{min}}}\right). \tag{17}$$

Proof. Samples from previous pdf: According to the variable jump distribution in (14), given k samples, the expected number of samples drawn from the previous posterior probability ($P(\theta|D^t)$) is $\lambda^t k$. By assumption, the algorithm has already drawn at least k samples in the previous algorithm time step. Therefore, by (8), the expected number of samples with distance less than ϵ from $\mathbb{E}(P_k)$ for samples drawn from previous distribution is at least $\lambda k \delta$

Samples from Gaussian: By (14), there are $k_0 = (1-\lambda^t)k$ samples drawn in expectation. According to (18), we have $Pr\Big\{\{P_k - \mathbb{E}(P_k)\} \leq \epsilon\Big\} \geq \delta_0$, where δ_0 is given by rearranging (8):

$$\delta_0 = 1 - 2e^{-2\epsilon^2 k_0}. \tag{18}$$

Thus, the expected number of samples with distance less than ϵ from $\mathbb{E}(P_k)$ is at least $\delta_0(1-\lambda^t)k$.

Overall reliability: The total expected number of samples with distance less than ϵ from $\mathbb{E}(P_k)$ is the sum of the above two parts. The overall reliability is obtained with dividing by k:

$$\delta_2 = \frac{(\lambda^t k \delta) + (\delta_0(1-\lambda^t)k)}{k} \tag{19}$$

Given the new obtained reliability, which is greater than the desire reliability helps us to decrease the number of evaluation. For the sake of illustration, Fig. 1 presents the minimum required number of evaluation with respect to λ for different precision and reliability. As it can be seen, the MCMC is equal to ARMCM if λ is always set to one. The number of evaluation in ARMCMC mitigates as the validity of the previous model increases. □

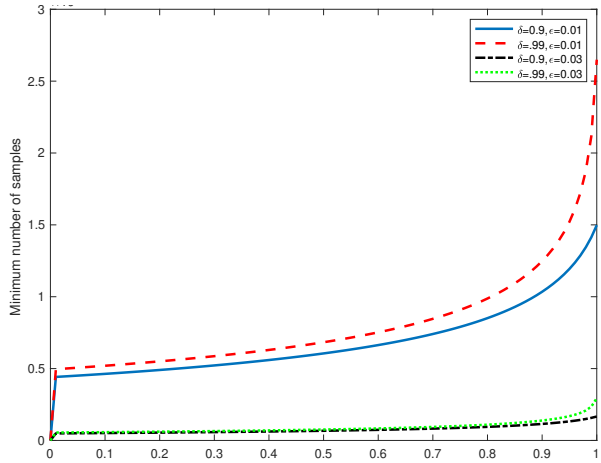

Figure 1: Minimum number of evaluation for different $\lambda, \epsilon, \delta$ in ARMCMC.

Table 1: Comparison of RLS Haddadi & Hashtrudi-Zaad (2012) and point estimate of ARMCMC and MCMC for environment identification.

ERRORS	MAE K_e	MAE B_e	MAE n	RMS F_e (MN)
RLS	0.5793	0.9642	0.3124	51.745
MCMC (N=1000)	0.6846	0.8392	0.3783	76.695
MCMC (T_s = 0.2)	0.7294	0.9964	0.4195	101.88
AR-APS	0.0774	0.0347	0.0945	33.774
AR-MAPS	0.0617	0.0316	0.0756	31.659

4 EMPIRICAL RESULTS

In this section, we demonstrate ARMCMC by identifying parameters of the Hunt-Crossley model, which represents an environment involving a needle contacting soft material. The needle is mounted as an end-effector on a high-precision translational robot, which switches between two modes: free motion and contact. Owing to the abrupt changes in model parameters when the contact is established or lost, online estimation of the force is extremely challenging.

4.1 RESULTS

The data sample time is $T = 1$ ms and each data pack includes 100 samples which results in an algorithm sample time equal to $T_s = 0.1$ s. All three parameters (K_e, B_e, p) prior knowledge are initialized with a prior of $N(1, 0.1)$ (Normal distribution with $\mu = 1$ and $\sigma = 0.1$) Moreover, more data is collected, the spread of the posterior pdf decreases. At A bit after 5 seconds, the needle goes outside of the soft material, and experiences zero force; this is equivalent to all parameters being set to zero. The color based visualization of probability distribution over time is used for three parameters in Fig. 2. During the period of time that the whole space is blue (zero probability density), there is no contact and the parameter values are equal to zero.

Since we are taking a Bayesian approach, we are able to estimate the entire posterior pdf. However, here for the sake of illustration, the point estimates are computed by averaging over the samples and shown in Fig. 3 for time-varying parameters $\theta_1 = K_e, \theta_1 = B_e, \theta_3 = p$. During the times that RLS results are chattering due to the use of saturation (if not, the results would have diverged), the needle is transitioning from being inside to outside of the soft material or vice versa. In addition, due to the assumption(12), performance can deteriorate even when there is no mode transition. Furthermore, in the RLS approach, estimated parameters suddenly diverged during free motion, since the regression vectors are linear dependent. In contrast, with the Bayesian approach, this issue can be easily resolved. The result of ARMCMC without EM (for delay compensation) is presented in Fig. 4, which shows force estimation with two different identification approaches. This probability of interest can be easily obtained by having derivation of parameter density at the disposal.

4.2 QUANTITATIVE COMPARISON

Quantitative details of comparing a naive point estimate of the ARMCMC algorithm by averaging over particles (AR-APS) and the RLS method are listed in Table 1. This reveals more than 70% improvement in the precision of all model parameters throughout the study using Mean Absolute Error (MAE) criteria and more than 57% improvements in the second norm of force estimation error. Among parameters, the viscose (B_e) has the largest error in RLS method since it is underestimated due to the restrictive assumption in equation 12. The point estimation obtained by considering the mode at modification phase and medium during reinforcement in ARMCMC is denoted as AR-MAPS. This approach uplifts the performance for parameter estimation for force estimation.

We also compare ARMCMC to MCMC. For the algorithm to run realtime, MCMC requires more time to converge. Two approaches to fix this is to reduce the number of samples, which results in worse precision and reliability compared to ARMCMC, or double the algorithm sample time which results in more delay in the estimation result, as well as slower respones to changes in the parameter.

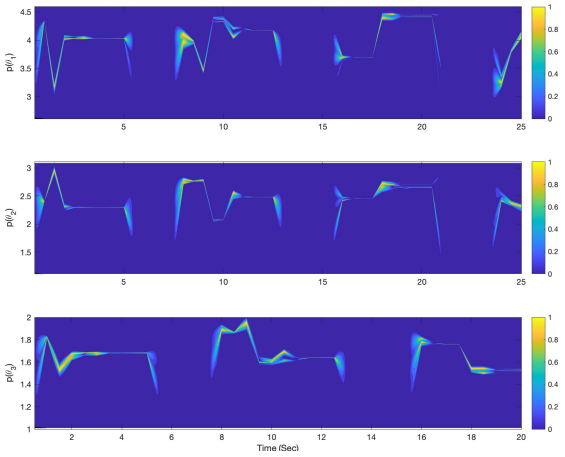

Figure 2: Probability distribution of (K_e, B_e, p) in ARMCMC.

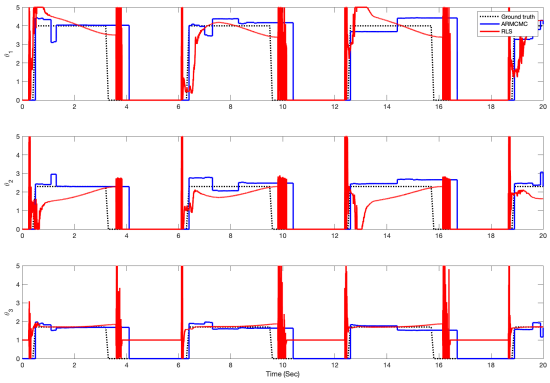

Figure 3: Environment parameter(K_e, B_e, p) point estimation for ARMCMC (AP-APS).

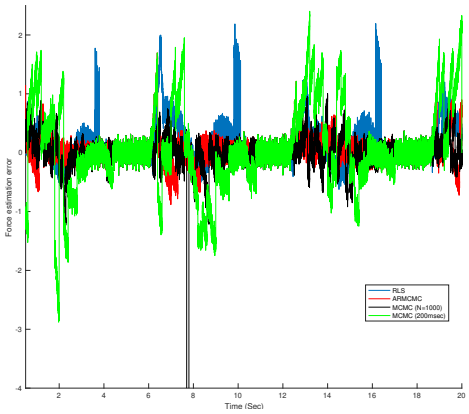

Figure 4: Force prediction error in RLS, ARMCMC, and MCMC

5 Conclusions

This paper presented an algorithm for online identification of full probability distribution of model parameters in a Bayesian paradigm using an adaptive recursive MCMC. Due to the abrupt change in parameters of models as such in contact with a soft environment when the contact is established/lost, conventional approaches suffer low performance. Empirical results on the Hunt-Crossley model as a nonlinear hybrid dynamic model comparing with a well known conventional identification process reveals proficiency of the proposed algorithm. The proposed method prove relaxing the prerequirement conditions besides the systematic strategy for handling abrupt changes in parameters. As future work, deploying a fully probabilistic framework from identification to control and decision-making stage is considered to exploit the full potentials of the Bayesian optimization.

REFERENCES

Niki Abolhassani, Rajni Patel, and Mehrdad Moallem. Needle insertion into soft tissue: A survey. *Medical Engineering and Physics*, 29(4):413–431, 2007.

Pedram Agand and Mahdi Aliyari Shoorehdeli. Adaptive model learning of neural networks with uub stability for robot dynamic estimation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6. IEEE, 2019.

Pedram Agand, Hamid D Taghirad, and Ali Khaki-Sedigh. Particle filters for non-gaussian hunt-crossley model of environment in bilateral teleoperation. In *4th International Conference on Robotics and Mechatronics (ICROM)*, pp. 512–517. IEEE, 2016.

Christopher M Bishop. Pattern recognition. *Machine Learning*, 128, 2006.

Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of markov chain monte carlo*. CRC press, 2011.

Noah J Cowan, Ken Goldberg, Gregory S Chirikjian, Gabor Fichtinger, Ron Alterovitz, Kyle B Reed, Vinutha Kallem, Wooram Park, Sarthak Misra, and Allison M Okamura. Robotic needle steering: Design, modeling, planning, and image guidance. In *Surgical Robotics*, pp. 557–582. Springer, 2011.

Nicola Diolaiti, Claudio Melchiorri, and Stefano Stramigioli. Contact impedance estimation for robotic systems. *IEEE Transactions on Robotics*, 21(5):925–935, 2005.

Peter L Green. Bayesian system identification of a nonlinear dynamical system using a novel variant of simulated annealing. *Mechanical Systems and Signal Processing*, 52:133–146, 2015.

Amir Haddadi and Keyvan Hashtrudi-Zaad. Real-time identification of hunt–crossley dynamic models of contact environments. *IEEE transactions on robotics*, 28(3):555–566, 2012.

KH Hunt and FRE Crossley. Coefficient of restitution interpreted as damping in vibroimpact. *Journal of applied mechanics*, 42(2):440–445, 1975.

Dominik Joho, Gian Diego Tipaldi, Nikolas Engelhard, Cyrill Stachniss, and Wolfram Burgard. Nonparametric bayesian models for unsupervised scene analysis and reconstruction. *Robotics: Science and Systems VIII*, pp. 161, 2013.

Shima Khatibisepehr, Biao Huang, and Swanand Khare. Design of inferential sensors in the process industry: A review of bayesian methods. *Journal of Process Control*, 23(10):1575–1596, 2013.

Scott Kuindersma, Roderic Grupen, and Andrew Barto. Variational bayesian optimization for run-time risk-sensitive control. *Robotics: Science and Systems VIII*, 2012.

Tomasz Kuśmierczyk, Joseph Sakaya, and Arto Klami. Variational bayesian decision-making for continuous utilities. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 6392–6402. Curran Associates, Inc., 2019.

Brett Ninness and Soren Henriksen. Bayesian system identification via markov chain monte carlo techniques. *Automatica*, 46(1):40–51, 2010.

Ali Noormohammadi-Asl and Hamid D Taghirad. Multi-goal motion planning using traveling salesman problem in belief space. *Information Sciences*, 471:164–184, 2019.

Harish Chaandar Ravichandar and Ashwin P Dani. Human intention inference using expectation-maximization algorithm with online model learning. *IEEE Trans. on Automation Science and Engineering*, 14(2):855–868, 2017.

Kaur Aare Saar, Fabio Giardina, and Fumiya Iida. Model-free design optimization of a hopping robot and its comparison with a human designer. *IEEE Robotics and Automation Letters*, 3(2): 1245–1251, 2018.

Roberto Tempo, Giuseppe Calafiore, and Fabrizio Dabbene. *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer Science & Business Media, 2012.

Felipe Tobar. Bayesian nonparametric spectral estimation. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 31*, pp. 10127–10137. Curran Associates, Inc., 2018.

Beilun Wang, Arshdeep Sekhon, and Yanjun Qi. A fast and scalable joint estimator for integrating additional knowledge in learning multiple related sparse Gaussian graphical models. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 5161–5170, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

Grady Williams, Brian Goldfain, Paul Drews, Kamil Saigol, J Rehg, and Evangelos A Theodorou. Robust sampling based model predictive control with sparse objective information. In *Robotics Science and Systems*, 2018.

A APPENDIX

Base on Abolhassani et al. (2007), a nonlinear hybrid model based on reality-based soft environment is considered as follows:

$$f_e = f_{\text{st}}(x, t, t_p) + f_{\text{fr}}(x, \dot{x}) + f_{\text{ct}}(x, t, t_p), \tag{20}$$

where x is the needle tip position and t_p is the latest time of puncture. Initial position of environment is assumed to be at the origin. The stiffness force (f_{st}) belongs to pre-puncture and the friction (f_{fr}) and cutting forces (f_{ct}) belong to post-puncture. The stiffness force is modeled using a nonlinear Hunt-Crossley model:

$$f_{st}(x, t, t_p) = \begin{cases} 0 & x < 0 \\ K_e x^p(t) & 0 \leq x \leq x_1, t < t_p \\ 0 & x > x_2, t \geq t_p \end{cases} \tag{21}$$

where K_e, p are the same parameter defined in (10). Maximum depth that the soft environment yields before puncture and its position after puncture is denoted by x_1, x_2, respectively ($0 < x_2 < x_1$). In this study, needle can insert up to 16.65, 10.21 mm before and after penetration. Friction model is inspired from modified Karnopp model.

$$f_{fr}(x, \dot{x}) = \begin{cases} C_n sgn(\dot{x}) + B_e x^p \dot{x} & \dot{x} \leq -\Delta v/2 \\ \max(D_n, F_a) & -\Delta v/2 < \dot{x} \leq 0 \\ \max(D_p, F_a) & 0 < \dot{x} < \Delta v/2 \\ C_p sgn(\dot{x}) + B_e x^p \dot{x} & \dot{x} \geq \Delta v/2 \end{cases} \tag{22}$$

where $C_n = -11.96 \times 10^{-3}$ and $C_p = 10.57 \times 10^{-3}$ are negative and positive values of dynamic friction, $D_n = -0.01823$ and $D_p = 0.01845$ are negative and positive values of static friction, B_e, p are same as Eq. (10). The relative velocity between the needle and tissue is denoted by $\dot{x}$, and $\Delta v/2 = 0.005$ is the value below which the velocity is considered to be zero, and F_a is the sum of non-frictional applied forces. The cutting force is considered as a static force constant for a tissue and needle geometry if soft tissues is not inhomogeneous and anisotropic Cowan et al. (2011)

$$f_{ct}(x, t, t_p) = \begin{cases} 0 & x \leq x_1, t < t_p \\ 0.94 & x > x_2, t \geq t_p \end{cases}. \tag{23}$$

According to the previous relations, the system is considered as a hybrid model while providing both free motion and contact environment. The manipulator is a translational mechanism with friction, slip, and hysteresis loop for the actuator. To present the superiority of the proposed algorithm, the results are compared with the RLS method presented in (Haddadi & Hashtrudi-Zaad, 2012). To prevent the results of RLS from divergence in model mismatch sequences, saturation is applied in the outputs of the identifier.