

Frequency Domain Forecasting Approach for Latency Reduction in Direct Human-Computer Interaction

Stanislav Aranovskiy, Rosane Ushirobira, Denis Efimov, Géry Casiez

► **To cite this version:**

Stanislav Aranovskiy, Rosane Ushirobira, Denis Efimov, Géry Casiez. Frequency Domain Forecasting Approach for Latency Reduction in Direct Human-Computer Interaction. CDC 2017 - 56th IEEE Conference on Decision and Control, Dec 2017, Melbourne, Australia. IEEE, Decision and Control (CDC), 2017 IEEE 56th Annual Conference on, pp.2623-2628, 2018, <10.1109/CDC.2017.8264040>. <hal-01651329>

HAL Id: hal-01651329

<https://hal.inria.fr/hal-01651329>

Submitted on 28 Nov 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Frequency Domain Forecasting Approach for Latency Reduction in Direct Human-Computer Interaction*

Stanislav Aranovskiy^{1,5}, Rosane Ushirobira², Denis Efimov^{2,3} and Géry Casiez^{3,4}

Abstract—The problem of latency reduction in direct human-computer interaction is considered and formulated as a trajectory prediction problem. To solve the problem, the predictor is constructed as a frequency-domain approximation of the non-causal ideal predictor. This approximation can be computed analytically, or obtained as an optimization task. An adaptive modification of the forecasting algorithm is proposed taking into account possible variations in user behavior. Experimental results illustrate the applicability of the proposed solution.

I. INTRODUCTION

In a human-computer interactive system, we define direct interaction as the co-location of the input information captured from the user and the related output information displayed to the user. Touchscreens are a typical example where the user fingers or a pen manipulates directly the objects of interest, write or draw. These objects are virtually attached to the touch points to mimic the manipulation of real world objects. However due to end-to-end latency, the manipulated objects “lag” behind the real pen or fingers.

For any type of interactive systems, the device requires some time to process the user input, to transfer it to the operating system, then to the specific application, to the graphical layer, and finally to display the output. This pipeline gives rise to the end-to-end latency, that is a time lag between the input action performed by the user and the reaction output displayed by the hardware. A general representation of this pipeline in direct touch interaction is given in Fig. 1. For modern touch-screen devices it is reasonable to expect the end-to-end latency between 60 ms and 200 ms as measured by [1].

Measuring the end-to-end latency and its variations is not a trivial task itself, and several studies were taken on this issue recently, for example in [2] and [3]. The detrimental impact of end-to-end latency on user performance has been known for a long time in indirect interactions with a computer mouse, see for example [4], and it is even more acute in direct touch interaction. In [5] the authors found that a latency greater than 25ms can significantly affect user performance in touch dragging tasks. At the same time, in

[1] the authors show that latency as small as 10ms can be perceived in direct touch interactions. Therefore, the problem of lag reduction for direct (and indirect) interactions is very challenging and important nowadays notably due to a widespread distribution of smartphones and tablet computers equipped with touch screens.

There are two ways to alleviate the impact of latency. The first is at the hardware level, *i.e.* to use more reactive and advanced elements and to make the signal flow as fast as possible. This approach has two drawbacks: the high cost of the advanced components and the increased energy consumption, which plays significant role for portable devices.

The second way to reduce the latency impact is at the software level and consists in using latency reduction algorithms. From the control point of view this problem can be formulated as a trajectory prediction or forecasting problem, and convenient prediction methods can be used. However, the methods based on underlying dynamic models can hardly be applied for latency reduction since dynamic models of user behavior are not typically available. There exist several dynamic models of human movements, *e.g.* the open-loop *minimum-jerk* optimal model used in [6], and the recently proposed dynamic models for indirect human-computer interaction [7], [8]. However, these models are valid for specific pointing movements only and cannot be applied for general direct interaction.

The lack of models motivates the use of model-free prediction methods. Particularly, trajectory prediction using Kalman filter for a chain of integrators was proposed in [9], and a method based on the first-order Taylor series was used in [10], where the velocity was estimated using the two most recent position measurements. In the recent paper [11] a forecasting algorithm based on the Taylor series expansion has been proposed, where the derivatives were estimated using either *algebraic* [12], or *homogeneous* finite-time [13] differentiators. The shortcoming is that, as it is stated in [11], high nonlinearity and interrelations between differentiator parameters make their tuning rather complicated.

Also some model-free approaches motivated by Kalman filter, curve fitting, and heuristic considerations can be found in the patents [14], [15] and [16], correspondingly.

In this paper we propose a novel model-free frequency-domain based approach. It can be shown that the ideal predictor is a *non-causal* linear time-invariant operator, which, obviously, cannot be implemented. However, assuming that human movements can be sufficiently well described by a finite number of low-frequency components, the imple-

*This work was supported by ANR (TurboTouch, ANR-14-CE24-0009). Stanislav Aranovskiy is supported by the Russian Science Foundation grant (project 17-79-20341).

¹ Equipe Automatique (AUT), CentraleSupélec – IETR, CS 47061 Avenue de la Boulaie, 35576 Cesson-Sévigné, France.

² Inria, Non-A team, 40 avenue Halley, 59650 Villeneuve d’Ascq, France.

³ CRISAL (UMR CNRS 9189), Villeneuve d’Ascq, France.

⁴ Inria, Mjolnir team, 40 avenue Halley, 59650 Villeneuve d’Ascq, France.

⁵ Department of Control Systems and Informatics, ITMO University, Kronverkskiy pr., 49, St. Petersburg, 197101, Russia

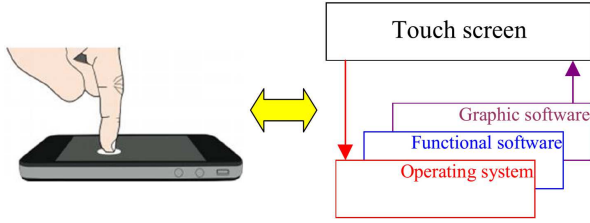


Fig. 1: Direct interaction diagram [11]

mentable forecasting algorithm is constructed as a *casual* low-frequency approximation of the ideal one. As it is shown in the paper, such an approximation can be either computed analytically, given *a priori* knowledge about the movement characteristics, or obtained numerically as an optimization task that can be efficiently solved given a sufficiently rich dataset. Moreover, an adaptive modification of the design is also proposed, which is supposed to adapt to changes of users and/or movement types. Experimental results illustrate the applicability of the proposed forecasting algorithms.

The rest of the paper is organized as follows. First, the problem statement is given in Section II. Next, the main forecasting algorithm is proposed in Section III, and its adaptive version is considered in Section IV. Experimental results are described in Section V, and conclusive remarks are given in Section VI.

II. PROBLEM STATEMENT

Consider a movement along a single axis. Motivated by physical reasoning, we assume that the trajectory $x(t)$ is smooth and bounded. The raw measurement $x(t_k)$ is the coordinate x measured at the time instance t_k , and integer $k \geq 0$. Given the latency value $t_L > 0$, the goal is to construct the forecasting algorithm

$$\hat{x}(t_k + t_L) = \mathcal{F}(\bar{x}(t_k), t_k),$$

where $\bar{x}(t_k) := [x(t_k), x(t_{k-1}), \dots, x(t_0)]^\top$, and $\hat{x}(t_k)$ is the estimate of $x(t_k)$. In other words, given the history $\bar{x}(t_k)$ up to the time instant t_k , we intend to forecast the *future* value $x(t_k + t_L)$. The forecasting algorithm $\mathcal{F}(\cdot)$ should be constructed in such a way that the forecasting error $e_x := \hat{x} - x$ is small in the sense of a certain norm.

We also assume that the coordinate x is measured with the constant sampling interval T_s and $t_k = kT_s$. Moreover, the latency value t_L is the integer number $L > 0$ of sampling intervals, $t_L = LT_s$ and $t_k + t_L = t_{k+L}$.

III. THE FORECASTING ALGORITHM

A. An ideal predictor approximation

The ideal predictor in the discrete-time domain is the L steps ahead time shift, *i.e.* $\mathcal{F}(\bar{x}(t_k), t_k) = x(t_{k+L}) = q^L x(t_k)$, where q is the one-step forward time shift operator. The ideal predictor has the following linear transfer function

$$F_{id}(z) = z^L,$$

where z is the complex variable. In frequency domain one obtains

$$F_{id}(i\omega) = e^{i\omega t_L},$$

where $i := \sqrt{-1}$ and ω is the frequency. It follows

$$\begin{aligned} |F_{id}(i\omega)| &\equiv 1, \\ \arg F_{id}(i\omega) &= \omega t_L. \end{aligned}$$

Obviously, such a transfer function is not casual and cannot be implemented. However, the concept of the ideal predictor motivates construction of the forecasting algorithm as an approximation of the ideal one in the frequency domain. Due to our assumptions on the trajectory $x(t)$, for any time instant t_k and any window size T_W , where $t_k > T_W > 0$, we can represent $x(t)$ for $t \in [t_k - T_W, t_k]$ as

$$x(t) = \sum_{i=0}^{\infty} A_i \sin(\omega_i t) + B_i \cos(\omega_i t), \quad (1)$$

where A_i and B_i are some constants, and

$$\omega_i = \frac{2\pi i}{T_W}.$$

Let us next assume that there exists the known value ω^\dagger , such that any reasonable trajectory generated by a user in human-computer interaction can be sufficiently well approximated with the first $N^\dagger + 1$ elements of the series expansion (1), where N^\dagger is chosen such that $\omega_{N^\dagger} \leq \omega^\dagger$ and $\omega_{N^\dagger+1} > \omega^\dagger$; this approximation is further denoted as x^\dagger ,

$$x^\dagger(t) = B_0 + \sum_{i=1}^{N^\dagger} A_i \sin(\omega_i t) + B_i \cos(\omega_i t). \quad (2)$$

Then the forecast \hat{x}^\dagger of this approximation can be used as the forecast of the coordinate x . The rest of the expansion (1)

$$\delta_x(t) := x(t) - x^\dagger(t) = \sum_{i=N^\dagger+1}^{\infty} A_i \sin(\omega_i t) + B_i \cos(\omega_i t)$$

is bounded and represents both negligible high-frequency trajectory deviations and possible measurement noise. Assuming that the coefficients of the approximation (2) are slow-varying (*i.e.* almost constant) during the latency interval t_L , the forecast \hat{x}^\dagger of the approximation x^\dagger is given by

$$\hat{x}^\dagger(t_k + t_L) = B_0 + \sum_{i=1}^{N^\dagger} A_i \sin(\omega_i(t_k + t_L)) + B_i \cos(\omega_i(t_k + t_L)).$$

This forecast can be constructed by applying the linear operator F_{ap} to the signal x^\dagger , where F_{ap} satisfies

$$F_{ap}(i\omega_i) = F_{id}(i\omega_i), \quad i \in \{0, 1, \dots, N^\dagger\}. \quad (3)$$

The operator F_{ap} is thus an approximation of the ideal predictor F_{id} : its frequency response coincides with the frequency response of F_{id} for frequencies $\omega_i \leq \omega^\dagger$, and does not necessarily coincide for others. Next, the frequency-domain approximation based forecasting algorithm takes the form

$$\hat{x}(t_k + t_L) = F_{ap}[x(t)] = \hat{x}^\dagger(t_k + t_L) + F_{ap}[\delta_x(t)], \quad (4)$$

where $F_{ap}[\cdot](t)$ denotes the operator F_{ap} applied to the signal $(\cdot)(t)$. The term $F_{ap}[\delta_x(t)]$ in the right-hand side of equation (4) represents the high-frequency distorting components and its impact should be minimized when possible.

B. Approximation using a Finite Impulse Response Filter

For any fixed ω^\dagger the operator F_{ap} can be chosen as a casual stable linear time-invariant system of order $N \geq 2N^\dagger$. One particular choice widely accepted in digital signal processing is the Finite Impulse Response (FIR) filter

$$F_{ap}(z) = c_0 + c_1 z^{-1} + \dots + c_N z^{-N} = \sum_{i=0}^N c_i z^{-i}$$

yielding the forecasting algorithm

$$\hat{x}(t_k + t_L) = c_0 x(t_k) + c_1 x(t_{k-1}) + \dots + c_N x(t_{k-N}).$$

Given $N^\dagger + 1$ different frequencies ω_i , $i = 0, 1, \dots, N^\dagger$, where

$$\omega_{N^\dagger} < \frac{\pi}{T_s}, \quad (5)$$

there always exists a unique FIR filter F_{ap} of order $N = 2N^\dagger$ satisfying (3), which can be constructed solving a system of the following $N + 1$ linear equations:

$$\begin{cases} \mathbf{1}_{N+1}^\top \bar{c} = 1, \\ \text{Re}(\Omega) \bar{c} = \text{Re}(\bar{F}_{id}), \\ \text{Im}(\Omega) \bar{c} = \text{Im}(\bar{F}_{id}), \end{cases} \quad (6)$$

where $\bar{c} := [c_0 \ c_1 \ \dots \ c_N]^\top$, $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ are the real and the imaginary parts of the complex value (\cdot) , respectively, $\mathbf{1}_{N+1}$ is the vector of 1's of dimension $N + 1$, the N^\dagger -dimensional vector with complex entries

$$\bar{F}_{id} := [e^{i\omega_1 t_L} \ e^{i\omega_2 t_L} \ \dots \ e^{i\omega_{N^\dagger} t_L}]^\top,$$

and the $N^\dagger \times (N + 1)$ matrix with complex entries

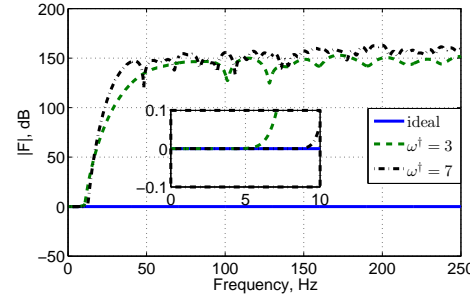
$$\Omega := \begin{bmatrix} 1 & e^{-i\omega_1 T_s} & e^{-i\omega_1 2T_s} & \dots & e^{-i\omega_1 N T_s} \\ 1 & e^{-i\omega_2 T_s} & e^{-i\omega_2 2T_s} & \dots & e^{-i\omega_2 N T_s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & e^{-i\omega_{N^\dagger} T_s} & e^{-i\omega_{N^\dagger} 2T_s} & \dots & e^{-i\omega_{N^\dagger} N T_s} \end{bmatrix}$$

is the complex generalization of the Vandermonde matrix. It can be shown (see proof of Theorem 1 in [17]) that the system (6) has its unique solution if¹ for all $i \in \{1, \dots, N^\dagger\}$ it holds $0 < \omega_i T_s < \pi$. The latter is satisfied due to (5).

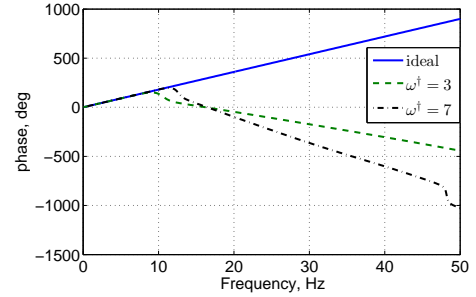
An example of frequency responses of such approximation FIR filters with $N = 2N^\dagger$ for $T_W = 2$, $L = 50$ and $T_s = 1$ ms is given in Fig. 2 for $\omega^\dagger = 3$ Hz, $N^\dagger = 38$ and $\omega^\dagger = 7$ Hz, $N^\dagger = 88$. Note the difference in the frequency axis limits.

As it can be seen from Fig. 2, the filter computed in this way has significant amplification for high frequencies, and it increases the impact of the $F_{ap}[\delta_x(t)]$ term, which can lead to unacceptable fast oscillations in the forecast \hat{x} , known as *jitter*. One possible way to overcome this drawback is to over-parametrize the FIR filter, *i.e.* to choose $N > 2N^\dagger$. Obviously, in such a case there exist infinitely many FIR

¹This condition is sufficient but not necessary.



(a) Magnitude (in dB) versus frequency (in Hz)



(b) Phase (in deg) versus frequency (in Hz)

Fig. 2: Comparison of the ideal predictor and its approximations given by FIR filters for $\omega^\dagger = 3$ Hz, $N^\dagger = 38$ and $\omega^\dagger = 7$ Hz, $N^\dagger = 88$.

filters satisfying the condition (3), and the predictor can be constructed via optimization procedure. For example, if no *a priori* information about δ_x is known, then it is reasonable to choose \bar{c} as

$$\bar{c} = \arg \min_{\bar{c}} \|F_{ap}(z)\|_\infty \text{ subject to (3).}$$

Otherwise, if some information about δ_x is available, *e.g.* its frequency range, the cost function may be modified to take this information into account.

C. Data-based filter tuning

The tuning procedure proposed in Subsection III-B requires *a priori* knowledge of the ω^\dagger value, as well as δ_x frequency range if available. However, these parameters are not easy to obtain: they might depend on the specific input device and thus are hardware-dependent. It makes the analytical tuning questionable and gives the rise to the experiment-based tuning approach.

Assume that a sufficiently large trajectory record \mathcal{R} is available representing all kinds of possible/admissible movements. The record \mathcal{R} consists of N_R measurements

$$\mathcal{R} := \{x(t_k), k = 1, 2, \dots, N_R\}.$$

Given the FIR filter order N the forecasting error for $k > L + N$ is computed as

$$\begin{aligned} e_x(t_k) &= \hat{x}(t_k) - x(t_k) = c_0 x(t_{k-L}) + c_1 x(t_{k-L-1}) \\ &\quad + \dots + c_N x(t_{k-L-N}) - x(t_k) = -x(t_k) + \sum_{i=0}^N c_i x(t_{k-L-i}). \end{aligned}$$

Define the vector of the forecasting errors as

$$\bar{e}_x := [e_x(t_{N_R}) \quad e_x(t_{N_R-1}) \quad \dots \quad e_x(t_{L+N+1})]^\top.$$

Then the FIR filter coefficients \bar{c} can be tuned in order to minimize a certain cost function $J(\bar{e}_x)$. Particularly, if the cost function is chosen as a vector norm

$$J(\bar{e}_x) = \|\bar{e}_x\|_p, \quad p = 1, 2, \infty,$$

then the tuning procedure is translated to the linear programming problem for $p = 1, \infty$ or to the least squares problem for $p = 2$, which can be efficiently solved even for a relatively large record size N_R , see [18], [19].

It is expected that the operator F_{ap} tuned with a sufficiently large record \mathcal{R} represents the best approximation of the ideal predictor F_{id} with respect to the ω^\dagger and δ_x information explicitly contained in the record.

IV. ADAPTIVE FORECAST

The approach considered in Subsections III-B and III-C proposes to design a single fixed-gain FIR filter and to use it for forecasting regardless the exact movement performed by a user. It is assumed that this fixed FIR filter is able to forecast sufficiently well for *all* possible trajectories and for *all* users. On the other hand, it is reasonable to assume that the ω^\dagger parameter can vary for different users and/or for different kinds of movements, *e.g.* for drawing or pointing. In this case the FIR filter tuned for a specific user and/or movement can provide better forecast accuracy for these specific conditions than a general fixed filter. In this section we propose to tune the FIR filter parameters on-line in order to adapt to the exact user and/or to the specific movement being executed at the moment.

A. Recursive least-squares algorithm

To this end we rewrite² the forecasting algorithm as

$$\hat{x}(k+L) = \phi^\top(k)\bar{c}(k), \quad (7)$$

where

$$\phi(k) := [x(k) \quad x(k-1) \quad \dots \quad x(k-N)]^\top \quad (8)$$

and $\bar{c}(k)$ is the vector of FIR filter coefficients updated at the each step. The goal is to update the coefficients $\bar{c}(k)$ in the way minimizing the cost function

$$J(k) = \sum_{i=L+N+1}^k \lambda^{k-i} \left(x(i) - \phi^\top(i-L)\bar{c}(k) \right)^2,$$

where $0 < \lambda \leq 1$ is the forgetting factor, which provides higher weight to recent measurements.

The desired update law is given with the recursive least-squares algorithm, see [20], [21]:

$$\begin{aligned} G(k) &= \frac{P(k-1)\phi(k-L)}{\lambda + \phi^\top(k-L)P(k-1)\phi(k-L)}, \\ P(k) &= \frac{1}{\lambda} \left(P(k-1) - G(k)\phi^\top(k-L)P(k-1) \right), \\ \bar{c}(k) &= \bar{c}(k-1) + G(k) \left(x(k) - \phi^\top(k-L)\bar{c}(k-1) \right). \end{aligned} \quad (9)$$

²To simplify notations in this section we will write $x(k)$ instead of $x(t_k)$.

At the each step k we get the new measurement $x(k)$ and update the gains \bar{c} using the earlier measurements $\phi(k-L)$. The updated gains are further used in the forecasting algorithm (7). The drawback of this approach is that the most recent measurements $x(k-1), \dots, x(k-L+1)$ are not actually used in the gains update. In other words, if any changes in movement have occurred, they do not impact the gains for the next L steps.

B. Adaptive predictor

To take into account the most recent measurements, the following procedure is proposed. First, we construct the one-step-ahead adaptive predictor. Next, the gains of this predictor are used to compute the gains \bar{c} . The one-step-ahead predictor is given by

$$\hat{x}(k+1) = \phi^\top(k)\theta(k), \quad (10)$$

where $\theta \in \mathbb{R}^{N+1}$ is the gains vector. The gains θ can be adaptively tuned using the recursive least square algorithm (9) and substituting $L = 1$ and θ in place of \bar{c} .

Next, to compute $\bar{c}(k)$ we assume that $\theta(k)$ is slow-varying for a time window of at least L steps ahead, and the trajectory x can be approximated on this window with the following *autoregressive* model with time-varying gains

$$x(k+1) = \phi^\top(k)\theta(k), \quad (11)$$

where for all $i \in \{0, 1, \dots, L\}$ it holds

$$\|\theta(k+i) - \theta(k)\| \leq \varepsilon_\theta, \quad (12)$$

and $\|\cdot\|$ is the Euclidean norm.

Proposition 1: Given the autoregressive model (11) under assumption (12), compute the vector $\bar{c}(k)$ as

$$\bar{c}(k) = [\theta(k) \quad \mathbf{e}_1 \quad \dots \quad \mathbf{e}_N]^\top \mathbf{e}_1, \quad (13)$$

where \mathbf{e}_i denotes the i -th Euclidean basis vector, *i.e.* the $N+1$ dimensional vector with a 1 in the i -th coordinate and 0's elsewhere. Then it holds

$$x(k+L) = \phi^\top(k)\bar{c}(k) + \sigma_x(k), \quad (14)$$

and

$$|\sigma_x(k)| \leq \sqrt{N+1}x_\infty H(\varepsilon_\theta, \|\theta(k)\|)\varepsilon_\theta, \quad (15)$$

where the hardware-dependent value x_∞ is the maximum admissible position value, the function

$$H(\varepsilon_\theta, \|\theta(k)\|) := \sum_{i=1}^{L-1} \binom{L-1}{i} \varepsilon_\theta^{i-1} (\|\theta(k)\| + 1)^{L-i}, \quad (16)$$

and $\binom{n}{k}$ is the binomial coefficient.

The proof of Proposition 1 is omitted due to space constrains.

The adaptive forecasting algorithm that uses the most recent measurements can be formulated as follows.

Algorithm 1.

- Step 1. At the k -th step get the new measurement $x(k)$.
- Step 2. Update the one-step-ahead predictor gains $\theta(k)$ using algorithm (9) with $L = 1$ and substituting θ in place of \bar{c} .



Fig. 3: Geometric shapes used in the experiment

Step 3. Compute $\bar{c}(k)$ as (13) using the one-step-ahead predictor gains $\theta(k)$.

Step 4. Compute forecast $\hat{x}(k+L)$ as (7).

V. EXPERIMENTAL RESULTS

In the considered experiment, three participants were asked to draw a set of figures divided into three groups: a) the geometric shapes given in Fig. 3, b) the Latin capital letters from A to F, and c) the digits from 0 to 9. The hardware used in the experiment is the Phantom Desktop [22], which is a pen-type haptic device. The Phantom Desktop is coupled with a horizontally oriented display screen and mimics a touchscreen with a pen. This setup allows to capture the measurements with the sampling interval $T_s = 1\text{ms}$. The latency value is estimated $t_L = 50\text{ms}$, that is $L = 50$. The movements are performed in the plane, and the x and y axes are considered separately.

We consider two forecasting algorithms: the fixed-gains FIR filter tuned according to the approach given in Subsection III-C, where the cost function is chosen as the Euclidean norm, *i.e.*

$$J(\bar{e}_x) = \|\bar{e}_x\|_2,$$

and the adaptive FIR filter given by *Algorithm 1*. Both FIR filters are of order $N = 15$, that implies 16 tunable gains. The forgetting factor is chosen as $\lambda = 0.9983$ implying that the most recent error value has double weight compared to the 400ms-old value. The frequency response of the fixed-gains FIR filter is given in Fig. 4. The phase response shows that the ω^\dagger value approximated from the given experimental data is close to 5Hz.

Both fixed-gains and adaptive forecasting algorithms perform well and are able to make a reasonable forecast for 50ms ahead, as it is validated with the considered experimental data. To evaluate the performance we use the following metrics:

- Mean Absolute Error (MAE),

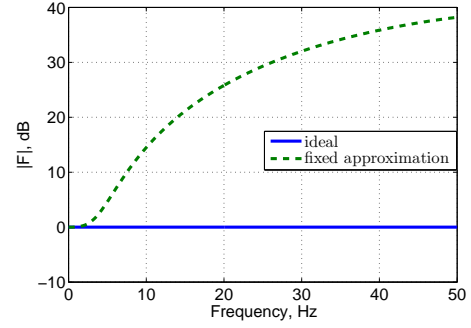
$$MAE := \frac{1}{N} \sum_{i=1}^N |e_x(t_k)|;$$

- Mean Squared Error (MAE),

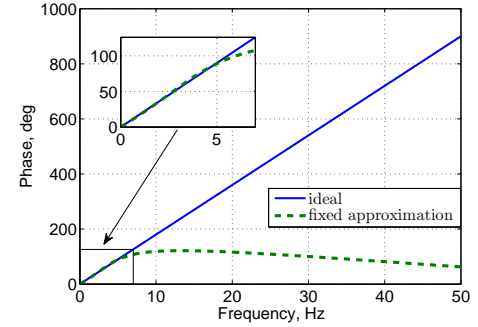
$$MSE := \frac{1}{N} \sum_{i=1}^N |e_x(t_k)|^2;$$

where N is the number of samples.

The resulting values of MAE and MSE computed over the whole experimental data for the fixed-gains FIR and for the adaptive FIR are given in Table I, and an example of trajectory forecast for x -axis is given in Fig. 5. The results illustrate that both fixed-gains and adaptive forecasts are



(a) Magnitude (in dB) versus frequency (in Hz)



(b) Phase (in deg) versus frequency (in Hz)

Fig. 4: Comparison of the ideal predictor and its approximations given by fixed-gains FIR filter tuned with $N = 15$

TABLE I: Numerical comparison of the fixed-gains and adaptive designs.

FIR type	MAE, pixels	MSE, pixels ²
Fixed-gains	2.35	12.0
Adaptive	2.40	14.7

close to the real position and are almost indistinguishable³. However, it should be highlighted that the fixed-gains FIR filter requires sufficiently rich records to be tuned off-line, while the adaptive FIR filter does not need such a preparation stage. Thus, the adaptive FIR filter is a solution with simpler tuning, where the price paid is a minor loss in performance.

VI. CONCLUSION

The problem of latency compensation in direct human-computer interaction is considered. This problem can be formulated as a forecasting algorithm design, and a novel approach is proposed based on frequency-domain approximation of the ideal predictor. Such an approximation can be either computed analytically or constructed solving optimization problem for a sufficiently rich dataset. Next, the adaptive modification of the forecasting algorithm is developed allowing possible variations of user behavior to be taken into account. Experimental studies illustrate the applicability of both fixed-gains and adaptive algorithms for trajectory forecasting.

³One probable reason behind this is that the movement characteristics, *i.e.* ω^\dagger and δ_x , are sufficiently homogeneous along the whole experiment.

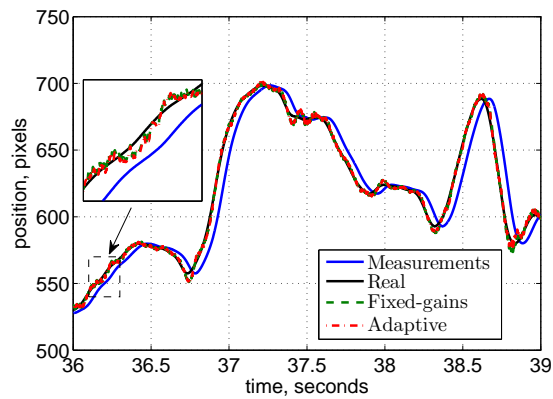


Fig. 5: An example of trajectory forecast for x-axis: measurements, real position, fixed-gains FIR filter forecast, and adaptive FIR filter forecast

One possible direction of further studies is to apply for adaptive parameters tuning some techniques with enhanced transient performance, *e.g.* the one recently proposed in [23]. This modification can increase forecasting accuracy at the moments when the user and/or trajectory type is changed. Another challenging problem is to consider model-based trajectory prediction algorithms that involves the design of user behavior models.

REFERENCES

- [1] A. Ng, J. Lepinski, D. Wigdor, S. Sanders, and P. Dietz, "Designing for low-latency direct-touch input," in *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 2012, pp. 453–464.
- [2] F. Bérard and R. Blanch, "Two touch system latency estimators: high accuracy and low overhead," in *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces*. ACM, 2013, pp. 241–250.
- [3] G. Casiez, S. Conversy, M. Falce, S. Huot, and N. Roussel, "Looking through the Eye of the Mouse : A Simple Method for Measuring End-to-end Latency using an Optical Mouse," in *Proceedings of the 28th ACM Symposium on User Interface Software and Technology*. ACM Press, 2015, pp. 629–636.
- [4] I. S. MacKenzie and C. Ware, "Lag as a determinant of human performance in interactive systems," in *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems*. ACM, 1993, pp. 488–493.
- [5] R. Jota, A. Ng, P. Dietz, and D. Wigdor, "How fast is fast enough?: a study of the effects of latency in direct-touch pointing tasks," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2013, pp. 2291–2300.
- [6] E. Lank, Y.-C. N. Cheng, and J. Ruiz, "Endpoint prediction using motion kinematics," in *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI 07)*, San Jose, California, USA, 2007, pp. 637–646.
- [7] P. Varnell and F. Zhang, "Characteristics of Human Pointing Motions with Acceleration," in *54th Conference on Decision and Control*, Osaka, Japan, 2015, pp. 5364–5369.
- [8] S. Aranovskiy, R. Ushirobira, D. Efimov, and G. Casiez, "Modeling pointing tasks in mouse-based human-computer interactions," in *55th Conference on Decision and Control*, Las-Vegas, USA, 2016, pp. 6595–6600.
- [9] J.-R. Wu and M. Ouhyoung, "On latency compensation and its effects on head-motion trajectories in virtual environments," *The visual computer*, vol. 16, no. 2, pp. 79–90, 2000.
- [10] E. Cattani, A. Rochet-Capellan, P. Perrier, and F. Bérard, "Reducing latency with a continuous prediction: Effects on users' performance in direct-touch target acquisitions," in *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces*. ACM, 2015, pp. 205–214.
- [11] R. Ushirobira, D. Efimov, G. Casiez, N. Roussel, and W. Perruquetti, "A forecasting algorithm for latency compensation in indirect human-computer interactions," in *European Control Conference*, Aalborg, Denmark, June 2016.
- [12] M. Mboup, C. Join, and M. Fliess, "Numerical differentiation with annihilators in noisy environment," *Numerical algorithms*, vol. 50, no. 4, pp. 439–467, 2009.
- [13] W. Perruquetti, T. Floquet, and E. Moulay, "Finite-time observers: application to secure communication," *IEEE Transactions on Automatic Control*, vol. 53, no. 1, pp. 356–360, 2008.
- [14] F. Moussavi, "Methods and apparatus for incremental prediction of input device motion," July 1 2014, uS Patent 8,766,915.
- [15] Qingkui, "Curve fitting based touch trajectory smoothing method and system," July 2 2014, cN Patent App. CN 201,210,585,264.
- [16] B. Kim and Y. Lim, "Mobile terminal and touch coordinate predicting method thereof," Aug. 28 2014, wO Patent App. PCT/KR2014/000,661.
- [17] S. Aranovskiy and L. B. Freidovich, "Adaptive compensation of disturbances formed as sums of sinusoidal signals with application to an active vibration control benchmark," *European Journal of Control*, vol. 19, no. 4, pp. 253–265, 2013.
- [18] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.
- [19] J. Currie and D. I. Wilson, "OPTI: Lowering the Barrier Between Open Source Optimizers and the Industrial MATLAB User," in *Foundations of Computer-Aided Process Operations*, N. Sahinidis and J. Pinto, Eds., Savannah, Georgia, USA, 8–11 January 2012.
- [20] S. Bittanti, P. Bolzern, and M. Campi, "Recursive least-squares identification algorithms with incomplete excitation: convergence analysis and application to adaptive control," *IEEE Transactions on Automatic Control*, vol. 35, no. 12, pp. 1371–1373, dec 1990.
- [21] K. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1994.
- [22] *PHANTOM Desktop specification*. [Online]. Available: <http://www.geomagic.com/archives/phantom-desktop/specifications/>
- [23] S. Aranovskiy, A. Bobtsov, R. Ortega, and A. Pyrkin, "Performance enhancement of parameter estimators via dynamic regressor extension and mixing," *IEEE Transactions on Automatic Control*, vol. PP, no. 99, pp. 1–1, 2016. [Online]. Available: <http://dx.doi.org/10.1109/TAC.2016.2614889>