



LUND UNIVERSITY

Identification of LTV Dynamical Models with Smooth or Discontinuous Time Evolution by means of Convex Optimization

Bagge Carlson, Fredrik; Robertsson, Anders; Johansson, Rolf

Published in:

2018 IEEE 14th International Conference on Control and Automation (ICCA)

DOI:

[10.1109/ICCA.2018.8444351](https://doi.org/10.1109/ICCA.2018.8444351)

2018

Document Version:

Early version, also known as pre-print

[Link to publication](#)

Citation for published version (APA):

Bagge Carlson, F., Robertsson, A., & Johansson, R. (2018). Identification of LTV Dynamical Models with Smooth or Discontinuous Time Evolution by means of Convex Optimization. In *2018 IEEE 14th International Conference on Control and Automation (ICCA)* IEEE - Institute of Electrical and Electronics Engineers Inc.. <https://doi.org/10.1109/ICCA.2018.8444351>

Total number of authors:

3

General rights

Unless other specific re-use rights are stated the following general rights apply:

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Read more about Creative commons licenses: <https://creativecommons.org/licenses/>

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

Identification of LTV Dynamical Models with Smooth or Discontinuous Time Evolution by means of Convex Optimization

Fredrik Bagge Carlson* Anders Robertsson Rolf Johansson

Abstract— We establish a connection between trend filtering and system identification which results in a family of new identification methods for linear, time-varying (LTV) dynamical models based on convex optimization. We demonstrate how the design of the cost function promotes a model with either a continuous change in dynamics over time, or causes discontinuous changes in model coefficients occurring at a finite (sparse) set of time instances. We further discuss the introduction of priors on the model parameters for situations where excitation is insufficient for identification. The identification problems are cast as convex optimization problems and are applicable to, e.g., ARX models and state-space models with time-varying parameters. We illustrate usage of the methods in simulations of jump-linear systems, a nonlinear robot arm with non-smooth friction and stiff contacts as well as in model-based, trajectory centric reinforcement learning on a smooth nonlinear system.

I. INTRODUCTION

The difficulty of the task of identifying time-varying dynamical models of systems varies greatly with the model considered and the availability of measurements of the state sequence. For smoothly changing dynamics, linear in the parameters, the recursive least-squares algorithm with exponential forgetting (RLS λ) is a common option. If a Gaussian random-walk model for the parameters is assumed, a Kalman filtering/smoothing algorithm [1] gives the filtering/smoothing densities of the parameters in closed form. The assumption of smoothly (Gaussian) varying dynamics is often restrictive. Discontinuous dynamics changes occur, for instance, when an external controller changes operation mode, when a sudden contact between a robot and its environment is established, an unmodeled disturbance enters the system or when a system is suddenly damaged.

Identification of systems with non-smooth dynamics evolution has been studied extensively. The book [2] treats the case where the dynamics are known, but the

state sequence unknown, i.e., state estimation. In [3], the authors examine the residuals from an initial constant dynamics fit to determine regions in time where improved fit is needed by the introduction of additional constant dynamics models. Results on identifiability and observability in jump-linear systems in the non-controlled (autonomous) setting are available in [4]. The main result on identifiability in [4] was a rank condition on a Hankel matrix constructed from the collected output data, similar to classical results on the least-squares identification of ARX models which appears as rank constraints on the, typically Toeplitz or block-Toeplitz, regressor matrix. Identifiability of the methods proposed in this article are discussed in Sec. V.

An LTV model can be seen as a first-order approximation of the dynamics of a nonlinear system around a trajectory. We emphasize that such an approximation will in general fail to generalize far from the this trajectory, but many methods in reinforcement learning and control make efficient use of the linearized dynamics for optimization, while ensuring validity of the approximation by constraints or penalty terms. An example provided in Sec. VIII highlights such a method.

An important class of identification methods that has been popularized lately is *trend filtering* methods [5], [6]. Trend filtering methods work by specifying a *fitness criterion* that determines the goodness of fit, as well as a *regularization* term, often chosen with sparsity promoting qualities. As a simple example, consider the reconstruction \hat{y} of a noisy signal $y = \{y_t \in \mathbb{R}\}_{t=1}^T$ with piecewise constant segments. To this end, we may formulate and solve the convex optimization problem

$$\underset{\hat{y}}{\text{minimize}} \quad \|y - \hat{y}\|_2^2 + \lambda \sum_t |\hat{y}_{t+1} - \hat{y}_t| \quad (1)$$

The first term is the fitness criterion or *loss function*, whereas the second term is a sparsity-promoting regularizer which promotes small changes between consecutive samples in the reconstructed signal. The sparsity promoting effect of the 1-norm regularizer is well known, and stems from the constant length of the gradient whenever the argument is non-zero [7]. Compare this to the squared difference, for which the gradient rapidly vanishes as the argument approaches zero. The squared difference will thus promote *small* arguments, whereas the 1-norm promotes *sparse* arguments.

*Open-source implementations of all presented methods and examples in this paper are made available at github.com/baggepinen/LTVModels.jl (Will be made available in advance of paper publication). The reported research was supported by the European Commission under the Framework Programme Horizon 2020 under grant agreement 644938 SARAFun. The authors are members of the LCCC Linnaeus Center and the eLLIIT Excellence Center at Lund University, Dept Automatic Control, Lund Sweden. Fredrik.Bagge_Carlson@control.lth.se

In this work, we will draw inspiration from the trend-filtering literature to develop new system identification methods for LTV models, with interesting properties. In trend filtering, we decompose a curve as a set of polynomial segments. In the identification methods proposed in this work, we instead decompose a multivariable state sequence as the output of a set of LTV models, where the model coefficients evolve as polynomial functions of time. We start by defining a set of optimization problems with a least-squares loss function and carefully chosen regularization terms. We further discuss how prior information can be utilized to increase the accuracy of the identification and end the article with identification of a nonlinear system with non-smooth friction and an example of model-based reinforcement learning followed by a discussion.

II. LTI IDENTIFICATION

We start by considering the case of identification of the parameters in an LTI model on the form

$$x_{t+1} = Ax_t + Bu_t + v_t, \quad t \in [1, T] \quad (2)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and input respectively. A discussion around the noise term v_t is deferred until Sec. IV-A, where we indicate how statistical assumptions on v_t influence the cost function and the properties of the estimate. If the state and input sequences are known, a plethora of methods for estimating the parameters exists. A common method for identification of systems that are linear in the parameters is the least-squares (LS) method, which in case of Gaussian noise, v , coincides with the maximum likelihood (ML) estimate. To facilitate estimation using the LS method, we write the model on the form $y = \Phi k$, and arrange the data according to

$$\begin{aligned} y &= \begin{bmatrix} x_1 \\ \vdots \\ x_T \end{bmatrix} && \in \mathbb{R}^{Tn} \\ k &= \text{vec} \left(\begin{bmatrix} A^\top & B^\top \end{bmatrix} \right) && \in \mathbb{R}^K \\ \Phi &= \begin{bmatrix} I_n \otimes x_0^\top & I_n \otimes u_0^\top \\ \vdots & \vdots \\ I_n \otimes x_{T-1}^\top & I_n \otimes u_{T-1}^\top \end{bmatrix} && \in \mathbb{R}^{Tn \times K} \end{aligned}$$

where \otimes denotes the Kronecker product and $K = n^2 + nm$ is the number of model parameters, and solve the optimization problem (3) with closed-form solution (4).

$$k^* = \arg \min_k \|\Phi k - y\|_2^2 \quad (3)$$

$$= (\Phi^\top \Phi)^{-1} \Phi^\top y \quad (4)$$

III. TIME-VARYING DYNAMICS

We now move on to the contribution of this work, and extend our view to systems where the dynamics change

with time. We limit the scope of this article to models on the form

$$\begin{aligned} x_{t+1} &= A_t x_t + B_t u_t + v_t \\ k_t &= \text{vec} \left(\begin{bmatrix} A_t^\top & B_t^\top \end{bmatrix} \right) \end{aligned} \quad (5)$$

where the parameters k are assumed to evolve according to the dynamical system

$$\begin{aligned} k_{t+1} &= H_t k_t + w_t \\ y_t &= (I_n \otimes [x_t^\top \quad u_t^\top]) k_t \end{aligned} \quad (6)$$

where, if no prior knowledge is available, the dynamics matrix H_t can be taken as the identity matrix; $H = I$ implies that the model coefficients follow a random walk dictated by the properties of w_t , i.e., the state transition density function $p_w(k_{t+1}|k_t)$. The emission density function $p_v(x_{t+1}|x_t, u_t, k_t)$ is determining the drift of the state, which for the parameter estimation problem can be seen as the distribution of measurements, given the current state of the system. We emphasize here that the state in the parameter evolution model refers to the current parameters k_t and not the system state x_t , hence, p_v is called the emission density and not the transition density. Particular choices of p_v and p_w emit data likelihoods concave in the parameters and hence amenable to convex optimization.

The following sections will introduce a number of optimization problems with different regularization functions, corresponding to different choices of p_w , and different regularization arguments, corresponding to different choices of H . We also discuss the quality of the identification resulting from the different modeling choices.

A. Low frequency time evolution

A slowly varying signal is characterized by *small first-order time differences*. To identify slowly varying dynamics parameters, we thus penalize the squared 2-norm of the first-order time difference of the model parameters, and solve the optimization problem

$$\underset{k}{\text{minimize}} \quad \|y - \hat{y}\|_2^2 + \lambda^2 \sum_t \|k_{t+1} - k_t\|_2^2 \quad (7)$$

where \sum_t denotes the sum over relevant indices t , in this case $t \in [1, T-1]$. This optimization problem has a closed form solution given by

$$\begin{aligned} \tilde{k}^* &= (\tilde{\Phi}^\top \tilde{\Phi} + \lambda^2 D_1^\top D_1)^{-1} \tilde{\Phi}^\top \tilde{Y} \\ \tilde{k} &= \text{vec}(k_1, \dots, k_T) \end{aligned} \quad (8)$$

where $\tilde{\Phi}$ and \tilde{Y} are appropriately constructed matrices and the first-order differentiation operator matrix D_1 is constructed such that $\lambda^2 \|D_1 \tilde{k}\|_2^2$ equals the second term in (7). The computational complexity $\mathcal{O}((TK)^3)$ of computing k^* using the closed-form solution (8) becomes prohibitive for all but toy problems. We note that the cost function in (7) is the negative data log-likelihood of a Brownian random-walk parameter model with $H = I$, which motivates us to develop a dynamic programming algorithm based on a Kalman smoother, detailed in Sec. IV-B.

B. Smooth time evolution

A smoothly varying signal is characterized by *small second-order time differences*. To identify smoothly time-varying dynamics parameters, we thus penalize the squared 2-norm of the second-order time difference of the model parameters, and solve the optimization problem

$$\underset{k}{\text{minimize}} \|y - \hat{y}\|_2^2 + \lambda^2 \sum_t \|k_{t+2} - 2k_{t+1} + k_t\|_2^2 \quad (9)$$

Also this optimization problem has a closed form solution on the form (8) with the corresponding second-order differentiation operator D_2 . Equation (9) is the negative data log-likelihood of a Brownian random-walk parameter model with added momentum and H derived in Sec. IV-C.2, where a Kalman smoother with augmented state is developed to find the optimal solution. We also extend problem (9) to more general regularization terms in Sec. IV-B.

C. Piecewise constant time evolution

In the presence of discontinuous or abrupt changes in the dynamics, estimation method (9) might perform poorly. A signal which is mostly flat, with a small number of distinct level changes, is characterized by a *sparse first-order time difference*. To detect sudden changes in dynamics, we thus formulate and solve the problem

$$\underset{k}{\text{minimize}} \|y - \hat{y}\|_2^2 + \lambda \sum_t \|k_{t+1} - k_t\|_2 \quad (10)$$

We can give (10) an interpretation as a *grouped-lasso* cost function, where instead of groups being formed out of variables, our groups are defined by differences between variables. We thus have a penalty on the 1-norm of the length of the difference vectors $k_{t+1} - k_t$ since $\|\|\cdot\|\|_1 = \|\|\cdot\|\|_2$. The 1-norm is a *sparsity-promoting* penalty, hence a solution in which only a small number of non-zero first-order time differences in the model parameters is favored, i.e., a piecewise constant dynamics evolution. At a first glance, one might consider the formulation

$$\underset{k}{\text{minimize}} \|y - \hat{y}\|_2^2 + \lambda \sum_t \|k_{t+1} - k_t\|_1 \quad (11)$$

which results in a dynamics evolution with sparse changes in the coefficients, but changes to different entries of k_t are not necessarily occurring at the same time instants. The formulation (10), however, promotes a solution in which the change occurs at the same time instants for all coefficients in A and B , i.e., $k_{t+1} = k_t$ for most t .

1) *Implementation*: Due to the non-squared norm penalty $\sum_t \|k_{t+1} - k_t\|_2$, problem (10) is significantly harder to solve than (9). An efficient implementation using the linearized ADMM algorithm [8] is made available in the accompanying repository.

TABLE I

SUMMARY OF OPTIMIZATION PROBLEM FORMULATIONS. D_n REFERS TO PARAMETER VECTOR TIME-DIFFERENTIATION OF ORDER n .

Norm	D_n	Result
1	1	Small number of steps (piecewise constant)
1	2	Small number of bends (piecewise affine)
2	1	Small steps (slowly varying)
2	2	Small bends (smooth)

D. Piecewise constant time evolution with known number of steps

If the number of switches in dynamics parameters, M , is known in advance, the optimal problem to solve is

$$\underset{k}{\text{minimize}} \|y - \hat{y}\|_2^2 \quad (12)$$

$$\text{subject to} \quad \sum_t \mathbf{1}\{k_{t+1} \neq k_t\} \leq M \quad (13)$$

where $\mathbf{1}\{\cdot\}$ is the indicator function. This problem is non-convex and we propose solving it using dynamic programming (DP). For this purpose we modify the algorithm developed in [9], an algorithm frequently referred to as segmented least-squares [10], where a curve is approximated by piecewise linear segments. The modification lies in the association of each segment (set of consecutive time indices during which the parameters are constant) with a dynamics model, as opposed to a simple straight line.¹ Unfortunately, the computational complexity of the dynamic programming solution, $\mathcal{O}(T^2K^3)$, becomes prohibitive for large T .²

E. Piecewise linear time evolution

A piecewise linear signal is characterized by a *sparse second-order time difference*, i.e., it has a small number of changes in the slope. A piecewise linear time-evolution of the dynamics parameters is hence obtained if we solve the optimization problem.

$$\underset{k}{\text{minimize}} \|y - \hat{y}\|_2^2 + \lambda \sum_t \|k_{t+2} - 2k_{t+1} + k_t\|_2 \quad (14)$$

F. Summary

The proposed optimization problems are summarized in Table I. The table illustrates how the choice of regularizer and order of time-differentiation of the parameter vector affects the quality of the resulting solution.

G. Two-step refinement

Since many of the proposed formulations of the optimization problem penalize the size of the changes to the parameters, solutions in which the changes are slightly underestimated are favored. To mitigate this issue, a two-step procedure can be implemented where in the first step,

¹Indeed, if a simple integrator is chosen as dynamics model and a constant input is assumed, the result of our extended algorithm reduces to the segmented least-squares solution.

²For details regarding the DP algorithm and implementation, the reader is referred to the source-code repository accompanying this article.

change points (knots) are identified. In the second step, the penalty on the one-norm is removed and equality constraints are introduced between consecutive time-indices for which no change in dynamics was indicated by the first step.

The second step can be computed very efficiently by noticing that the problem can be split into several identical sub-problems at the knots identified in the first step. The sub-problems have closed-form solutions if the problem in Sec. III-C is considered.

To identify the points at which the dynamics change, we observe the argument inside the sum of the regularization term, i.e., $a_{t1} = \|k_{t+1} - k_t\|_2$ or $a_{t2} = \|k_{t+2} - 2k_{t+1} + k_t\|_2$. Time instances where a_t is taking non-zero values indicate change points.

IV. DYNAMICS PRIOR AND KALMAN FILTERING

The identifiability of the parameters in a dynamical model hinges on the observability of the dynamics system (6), or more explicitly, only modes excited by the input u will be satisfactorily identified. If the identification is part of an iterative learning and control scheme, e.g., ILC or reinforcement learning, it might be undesirable to introduce additional noise in the input to improve excitation for identification. This section will introduce prior information about the dynamics which mitigates the issue of poor excitation of the system modes. The prior information might come from, e.g., a nominal model known to be inaccurate, or an estimated global model such as a Gaussian mixture model (GMM). A statistical model of the joint density $p(x_{t+1}, x_t, u_t)$ constructed from previously collected tuples (x_{t+1}, x_t, u_t) provides a dynamical model of the system through the conditional pdf $p(x_{t+1}|x_t, u_t)$.

We will see that for priors from certain families, the resulting optimization problem remains convex. For the special case of a Gaussian prior over the dynamics parameters or the output, the posterior mean of the parameter vector is conveniently obtained from a Kalman-smoothing algorithm, modified to include the prior.

A. General case

If we introduce a parameter state k (c.f., (5)) and a prior over all parameter-state variables $p(k_t|z_t)$, where the variable z_t might be, for instance, the time index t or state x_t , we have the data log-likelihood

$$\begin{aligned} \log p(k, y|x, z)_{1:T} &= \sum_{t=1}^T \log p(y_t|k_t, x_t) \\ &+ \sum_{t=1}^{T-1} \log p(k_{t+1}|k_t) + \sum_{t=1}^T \log p(k_t|z_t) \end{aligned} \quad (15)$$

which factors conveniently due to the Markov property of a state-space model. For particular choices of density functions in (15), notably Gaussian and Laplacian, the negative likelihood function becomes convex. The next section will elaborate on the Gaussian case and introduce

a recursive algorithm that solves for the full posterior efficiently. The Laplacian case, while convex, does not admit an equally efficient algorithm, but is more robust to outliers in the data.

B. Gaussian case

If all densities in (15) are Gaussian and k is modeled with the Brownian random walk model (6) (Gaussian v_t), (15) can be written on the form (scaling constants omitted)

$$\begin{aligned} -\log p(k, y|x, z)_{1:T} &= \sum_{t=1}^T \|y_t - \hat{y}(k_t, x_t)\|_{\Sigma_y^{-1}}^2 \\ &+ \sum_{t=1}^{T-1} \|k_{t+1} - k_t\|_{\Sigma_k}^2 \\ &+ \sum_{t=1}^T \|\mu_0(z_t) - k_t\|_{\Sigma_0^{-1}(z_t)}^2 \end{aligned} \quad (16)$$

for some function $\mu_0(z_t)$ which produces the prior mean of k given z_t . $\Sigma_y, \Sigma_k, \Sigma_0(z_t)$ are the covariance matrices of the state-drift, parameter drift and prior respectively and $\|x\|_{\Sigma^{-1}}^2 = x^T \Sigma^{-1} x$.

In this special case, we introduce a recursive solution given by a modified Kalman smoothing algorithm, where the conditional mean of the state is updated with the prior. Consider the standard Kalman filtering equations, reproduced here to establish the notation

$$\hat{x}_{t|t-1} = A\hat{x}_{t-1|t-1} + Bu_{t-1} \quad (17)$$

$$P_{t|t-1} = AP_{t-1|t-1}A^T + R_1 \quad (18)$$

$$K_t = P_{t|t-1}C^T(CP_{t|t-1}C^T + R_2)^{-1} \quad (19)$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t(y_t - C\hat{x}_{t|t-1}) \quad (20)$$

$$P_{t|t} = P_{t|t-1} - K_tCP_{t|t-1} \quad (21)$$

where x is the state vector, with state-drift covariance R_1 and C is a matrix that relates x to a measurement $y = Cx$ with covariance R_2 . The first two equations constitute the *prediction* step, and the last two equations incorporate the measurement y_t in the *correction* step. The modification required to incorporate a Gaussian prior on the state variable $p(x_t|v_t) = \mathcal{N}(\mu_0(v_t), \Sigma_0(v_t))$ involves a repeated correction step and takes the form

$$\bar{K}_t = P_{t|t}(P_{t|t} + \Sigma_0(v_t))^{-1} \quad (22)$$

$$\bar{x}_{t|t} = \hat{x}_{t|t} + \bar{K}_t(\mu_0(v_t) - \hat{x}_{t|t}) \quad (23)$$

$$\bar{P}_{t|t} = P_{t|t} - \bar{K}_tP_{t|t} \quad (24)$$

where $\bar{\cdot}$ denotes the posterior value. This additional correction can be interpreted as receiving a second measurement $\mu_0(v_t)$ with covariance $\Sigma_0(v_t)$. For the Kalman-smoothing algorithm, $\hat{x}_{t|t}$ and $P_{t|t}$ in (23) and (24) are replaced with $\hat{x}_{t|T}$ and $P_{t|T}$.

A prior over the output of the system, or a subset thereof, is straight forward to include in the estimation by means of an extra update step, with C, R_2 and y being replaced with their corresponding values according to the prior.

C. Kalman filter for identification

We can employ the Kalman-based algorithm to solve two of the proposed optimization problems:

1) *Low frequency*: The Kalman smoother can be used for solving identification problems like (7) by noting that (7) is the negative log-likelihood of the dynamics model (6). The identification problem is thus reduced to a standard state-estimation problem.

2) *Smooth*: To develop a Kalman-filter based algorithm for solving (9), we augment the model (6) with the state variable $k'_t = k_t - k_{t-1}$ and note that $k'_{t+1} - k'_t = k_{t+1} - 2k_t + k_{t-1}$. We thus introduce the augmented-state model

$$\begin{bmatrix} k_{t+1} \\ k'_{t+1} \end{bmatrix} = \begin{bmatrix} I_K & I_K \\ 0_K & I_K \end{bmatrix} \begin{bmatrix} k_t \\ k'_t \end{bmatrix} + \begin{bmatrix} 0 \\ w_t \end{bmatrix} \quad (25)$$

$$y_t = [(I_n \otimes [x_t^\top \ u_t^\top]) \ 0] \begin{bmatrix} k_t \\ k'_t \end{bmatrix} \quad (26)$$

which is on a form suitable for filtering/smoothing with the machinery developed above.

3) *General case*: The Kalman-filter based identification method can be generalized to solving optimization problems where the argument in the regularizer appearing in (9) is replaced by a general linear operation on the parameter vector, $P(z)k$, and we have the following proposition

Proposition 1: Any optimization problem on the form

$$\underset{k}{\text{minimize}} \|y - \hat{y}\|_2^2 + \lambda^2 \sum_t \|P(z)k_t\|_2^2 \quad (27)$$

where $P(z)$ is a polynomial of degree $n > 0$ in the time difference operator z with $z^{-n}P(1) = 0$, can be solved with a Kalman smoother employed to an autonomous state-space system.

Proof: Let $P^*(z^{-1}) = z^{-n}P(z)$. We assume without loss of generality that $P^*(0) = 1$ since any constant $P^*(0)$ can be factored out of the polynomial. $Q(z^{-1}) = P^*(z^{-1})^{-1}$ is a strictly proper transfer function and has a realization as a linear, Gaussian state-space system of degree n . Since $Q(z^{-1})$ is strictly proper, the realization has no direct term. The negative data log-likelihood of $Q(z^{-1})$ is equal, up to constants independent of k , to the cost function in Eq. (27), hence the Kalman smoother applied to Q optimizes Eq. (27). ■

For (9) $P(z)$ equals $z^2 - 2z + 1$ and $Q(z^{-1})$ has a realization on the form (25).

V. WELL-POSEDNESS AND IDENTIFIABILITY

To assess the well-posedness of the proposed identification methods, we start by noting that the problem of finding A in $x_{t+1} = Ax_t$ given a pair (x_{t+1}, x_t) is an ill-posed problem in the sense that the solution is non unique. If we are given several pairs (x_{t+1}, x_t) , for different t , while A remains constant, the problem becomes over-determined and well-posed in the least-squares sense, provided that the vectors $\{x_t^{(i)}\}_{i=1}^T$ span \mathbb{R}^n . The LTI-case in Sec. II is well posed according to classical results, when Φ has full column rank.

When we extend our view to LTV models, the number of free parameters is increased significantly, and the corresponding regressor matrix $\tilde{\Phi}$ will never have full column rank and the introduction of a regularization term is necessary. Informally, for every n measurements, we have $K = n^2 + nm$ free parameters. If we consider the identification problem of Eq. (10) and let $\lambda \rightarrow \infty$, the regularizer terms essentially becomes equality constraints. This will enforce a solution in which all parameters in k are constant over time, and the problem reduces to the LTI-problem. As λ decreases, the effective number of free parameters increases until the problem gets ill-posed for $\lambda = 0$. We formalize the above arguments as

Proposition 2: Optimization problems (7) and (10) have unique global minima for $\lambda > 0$ if and only if the corresponding LTI optimization problem has a unique solution.

Proof: The cost function is a sum of two convex terms. For a global minimum to be non-unique, the Hessians of the two terms must have intersecting nullspaces. In the limit $\lambda \rightarrow \infty$ the problem reduces to the LTI problem. The nullspace of the regularization Hessian, which is invariant to λ , does thus not share any directions with the nullspace of $\tilde{\Phi}^\top \tilde{\Phi}$ which establishes the equivalence of identifiability between the LTI problem and the LTV problems. ■

Proposition 3: Optimization problems (9) and (14) with higher order differentiation in the regularization term have unique global minima for $\lambda > 0$ if and only if there exists no vector $v \in \mathbb{R}^{n+m}$ such that

$$C_t^{xu} v = \begin{bmatrix} x_t x_t^\top & x_t u_t^\top \\ u_t x_t^\top & u_t u_t^\top \end{bmatrix} v = 0 \quad \forall t \quad (28)$$

Proof: Again, the cost function is a sum of two convex terms and for a global minimum to be non-unique, the Hessians of the two terms must have intersecting nullspaces. In the limit $\lambda \rightarrow \infty$ the regularization term reduces to a linear constraint set, allowing only parameter vectors that lie along a line through time. Let $\tilde{v} \neq 0$ be such a vector, parametrized by t as $\tilde{v} = [\tilde{v}^\top \ 2\tilde{v}^\top \ \dots \ T\tilde{v}^\top]^\top \in \mathbb{R}^{TK}$ where $\tilde{v} = \text{vec}(\{v\}_1^N) \in \mathbb{R}^K$ and v is an arbitrary vector $\in \mathbb{R}^{n+m}$. $\tilde{v} \in \text{null}(\tilde{\Phi}^\top \tilde{\Phi})$ implies that the loss is invariant to the perturbation $\alpha \tilde{v}$ to \tilde{k} for an arbitrary $\alpha \in \mathbb{R}$. $(\tilde{\Phi}^\top \tilde{\Phi})$ is given by $\text{blkdiag}(\{I_n \otimes C_t^{xu}\}_1^T)$ which means that $\tilde{v} \in \text{null}(\tilde{\Phi}^\top \tilde{\Phi}) \iff \alpha t (I_n \otimes C_t^{xu}) \tilde{v} = 0 \quad \forall (\alpha, t) \iff \tilde{v} \in \text{null}(I_n \otimes C_t^{xu}) \quad \forall t$, which implies $v \in \text{null}(C_t^{xu})$ due to the block-diagonal nature of $I_n \otimes C_t^{xu}$. ■

For the LTI problem to be well-posed, the system must be identifiable and the input u must be persistently exciting of sufficient order [11].

VI. EXAMPLE – JUMP-LINEAR SYSTEM

We now consider a simulated example. We generate a state sequence from the following LTV system, where the change in dynamics, from

$$A_t = \begin{bmatrix} 0.95 & 0.1 \\ 0.0 & 0.95 \end{bmatrix}, \quad B_t = \begin{bmatrix} 0.2 \\ 1.0 \end{bmatrix}$$

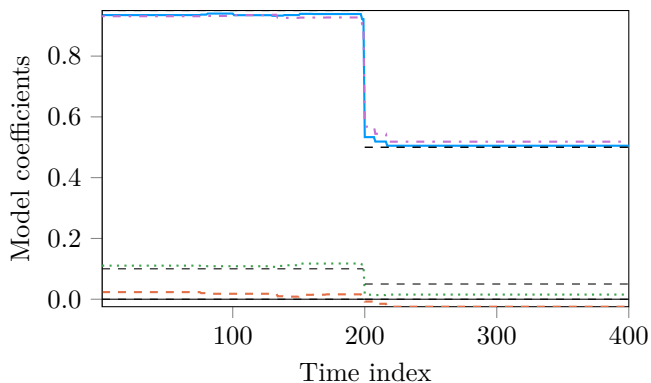


Fig. 1. Piecewise constant state-space dynamics. True values are shown with dashed, black lines. Gaussian state-transition and measurement noise with $\sigma = 0.2$ were added.

to

$$A_t = \begin{bmatrix} 0.5 & 0.05 \\ 0.0 & 0.5 \end{bmatrix}, \quad B_t = \begin{bmatrix} 0.2 \\ 1.0 \end{bmatrix}$$

occurred at $t = 200$. The input was Gaussian noise of zero mean and unit variance, state transition noise and measurement noise of zero mean and $\sigma = 0.2$ were added. Figure 1 depicts the estimated coefficients in the dynamics matrices for a value of λ chosen using the L-curve method [12].

VII. EXAMPLE – NON-SMOOTH ROBOT ARM WITH STIFF CONTACT

To illustrate the ability of the proposed models to represent the non-smooth dynamics along a trajectory of a robot arm, we simulate a two-link robot with discontinuous Coulomb friction. We also let the robot establish a stiff contact with the environment to illustrate both strengths and weaknesses of the modeling approach.

The state of the robot arm consists of two joint coordinates, q , and their time derivatives, \dot{q} . Figure 2 illustrates the state trajectories, control torques and simulations of a model estimated by solving (10). The figure clearly illustrates that the model is able to capture the dynamics both during the non-smooth sign change of the velocity, but also during establishment of the stiff contact. The learned dynamics of the contact is however time-dependent, which is illustrated in Figure 3, where the model is used on a validation trajectory where a different noise sequence was added to the control torque. Due to the novel input signal, the contact is established at a different time-instant and as a consequence, there is an error transient in the simulated data.

VIII. EXAMPLE – REINFORCEMENT LEARNING

In this example, we use the proposed methods to identify LTV dynamics models for reinforcement learning. The goal of the task is to dampen oscillations of a pendulum attached to a moving cart by means of moving the cart, with bounds on the control signal and a quadratic cost on states and control. Due to the nonlinear nature of the

pendulum dynamics, linear expansions of the dynamics in the upward (initial) position and downward (final) position have poles on opposite sides of the imaginary axis. To this end, we employ a reinforcement-learning framework inspired by [13], where we perform a series of rollouts whereafter each we 1) fit a dynamics model along the last obtained trajectory, 2) optimize the cost function under the model using iterative LQG (differential dynamic programming),³ an algorithm that calculates the value function exactly under the LTV dynamics and a quadratic expansion of the cost function. In order to stay close to the validity region of the linear model, we put bounds on the deviation between each new trajectory and the last trajectory. We compare three different models; the ground truth system model, an LTV model (obtained by solving (9)) and an LTI model. The total cost over $T = 500$ time steps is shown as a function of learning iteration in Fig. 4. The figure illustrates how the learning procedure reaches the optimal cost of the ground truth model when an LTV model is used, whereas when using an LTI model, the learning diverges. The figure further illustrates that if the LTV model is fit using a prior (Sec. IV-B), the learning speed is increased. The prior in this case was constructed from the true system model, linearized around the last trajectory. This strategy is unavailable in a real application, but the experiment serves as an indication of the effectiveness of inclusion of a prior in this example. Future work is targeting the incremental estimation of these priors.

IX. DISCUSSION

This article presents methods for estimation of linear, time-varying models. The methods presented extend directly to nonlinear models that remain *linear in the parameters*. When estimating an LTV model from a trajectory obtained from a nonlinear system, one is effectively estimating the linearization of the system around that trajectory. A first-order approximation to a nonlinear system is not guaranteed to generalize well as deviations from the trajectory become large. Many nonlinear systems are, however, approximately *locally* linear, such that they are well described by a linear model in a small neighborhood around the linearization/operating point. For certain methods, such as iterative learning control and trajectory centric reinforcement learning, a first-order approximation to the dynamics is used for efficient optimization, while the validity of the approximation is ensured by incorporating penalties or constraints between two consecutive trajectories.

The methods presented allow very efficient learning of this first-order approximation due to the prior belief over the nature of the change in dynamics parameters, encoded by the regularization terms. By postulating a prior belief that the dynamics parameters change in a certain way, less demand is put on the data required

³Implementation made available at github.com/baggepinnen/DifferentialDynamicProgramming.jl

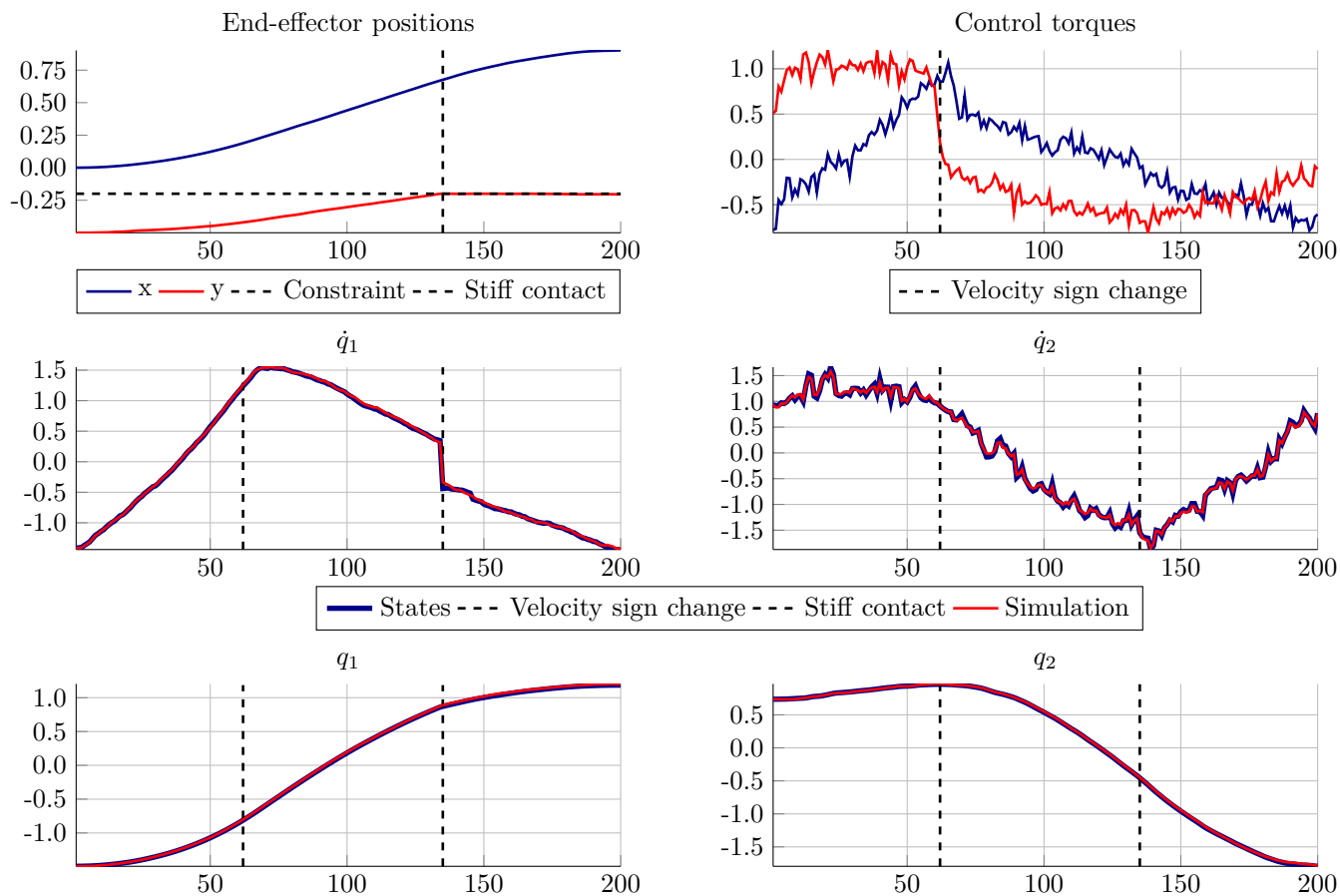


Fig. 2. Simulation of non-smooth robot dynamics with stiff contact – training data vs. sample time index. The sign change in velocity, and hence a discontinuous change in friction torque, occurs in the time interval 50-100 and the contact is established in the time interval 100-150. For numerical stability, all time-series are normalized to zero mean and unit variance, hence, the original velocity zero crossing is explicitly marked with a dashed line. The control signal plot clearly indicates the discontinuity in torque around the unnormalized zero crossing of \dot{q}_2 .

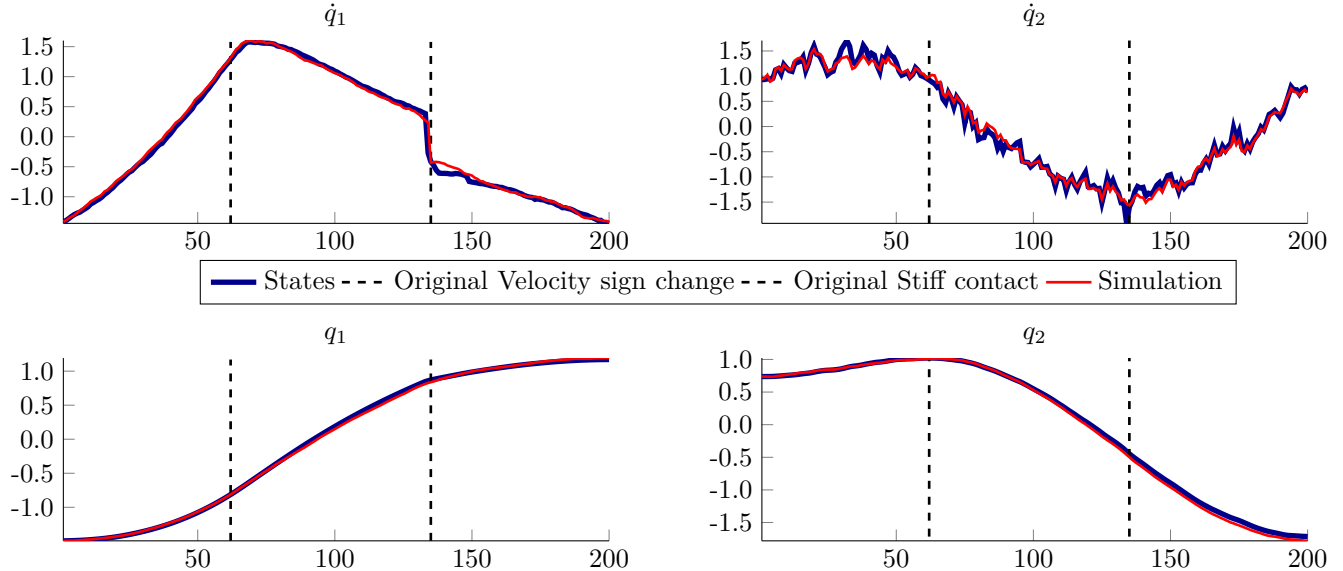


Fig. 3. Simulation of non-smooth robot dynamics with stiff contact – validation data vs. sample time index. The dashed lines indicate the event times for the training data, highlighting that the model is able to deal effortlessly with the non-smooth friction, but inaccurately predicts the time evolution around the contact event which now occurs at a slightly different time instance.

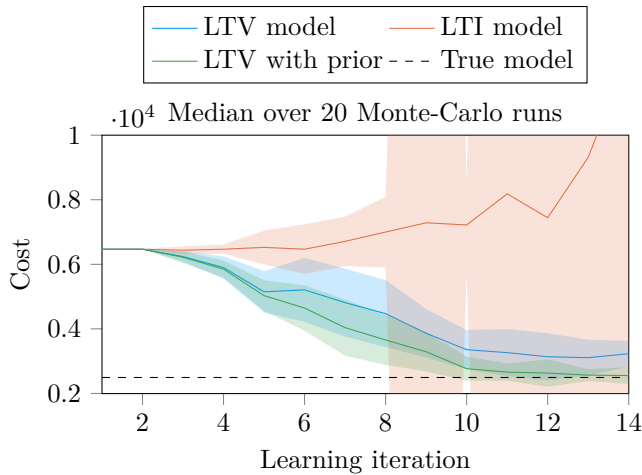


Fig. 4. Reinforcement learning example. Three different model types are used to iteratively optimize the trajectory of a pendulum on a cart. Due to the nonlinear nature of the pendulum dynamics, linear expansions of the dynamics in the upward and downward positions have poles on opposite sides of the imaginary axis, why the algorithm fails with an LTI model.

for identification. The identification process will thus not interfere with normal operation in the same way as if excessive noise would be added to the input for identification purposes. This allows learning of flexible, over-parametrized models that fit available data well. This makes the proposed identification methods attractive in applications such as guided policy search (GPS) [13], [14] and non-linear iterative learning control (ILC) [15], where they can lead to dramatically decreased sample complexity.

When faced with a system where time-varying dynamics is suspected and no particular knowledge regarding the dynamics evolution is available, or when the dynamics are known to vary slowly, a reasonable first choice of algorithm is (9). It is also by far the fastest of the proposed methods due to the Kalman-filter implementation of Sec. IV-B.⁴ Example use cases include when dynamics are changing with a continuous auxiliary variable, such as temperature, altitude or velocity. If a smooth parameter drift is found to correlate with an auxiliary variable, LPV-methodology can be employed to model the dependence explicitly.

Dynamics may change abruptly as a result of, e.g., system failure, change of operating mode, or when a sudden disturbance enters the system, such as a policy change affecting a market or a window opening affecting the indoor temperature. The identification method (10) can be employed to identify when such changes occur, without specifying a priori how many changes are expected.

For simplicity, the regularization weights were kept as simple scalars in this article. However, all terms $\lambda \|\Delta k\|_2^2 = (\Delta k)^T (\lambda I) (\Delta k)$ can be generalized to

$(\Delta k)^T \Lambda (\Delta k)$, where Λ is an arbitrary positive definite matrix. This allows incorporation of different scales for different variables with little added implementation complexity.

X. CONCLUSIONS

We have proposed a framework for identification of linear, time-varying models along trajectories of nonlinear dynamical systems using convex optimization. We showed how a Kalman smoother can be used to estimate the dynamics efficiently in a few special cases, and demonstrated the use of the proposed LTV models on three examples, highlighting their efficiency for trajectory-centric, model-based reinforcement learning, iterative learning control (ILC), and jump-linear system identification. We have also demonstrated the ability of the models to handle non-smooth friction dynamics as well as analyzed the identifiability of the models.

REFERENCES

- [1] H. E. Rauch, F. Tung, C. T. Striebel, *et al.*, “Maximum likelihood estimates of linear dynamic systems,” *AIAA journal*, vol. 3, no. 8, 1965.
- [2] O. L. V. Costa, M. D. Fragoso, and R. P. Marques, *Discrete-time Markov jump linear systems*. Springer Science & Business Media, London, 2006.
- [3] S. Nagarajaiah and Z. Li, “Time segmented least squares identification of base isolated buildings,” *Soil Dynamics and Earthquake Engineering*, vol. 24, no. 8, 2004.
- [4] R. Vidal, A. Chiuso, and S. Soatto, “Observability and identifiability of jump linear systems,” in *IEEE Conf. Decision and Control (CDC), Las Vegas*, vol. 4. IEEE, 2002.
- [5] S.-J. Kim, K. Koh, S. Boyd, and D. Gorinevsky, “ ℓ_1 trend filtering,” *SIAM review*, vol. 51, no. 2, 2009.
- [6] R. J. Tibshirani *et al.*, “Adaptive piecewise polynomial estimation via trend filtering,” *The Annals of Statistics*, vol. 42, no. 1, 2014.
- [7] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, Cambridge, Massachusetts, 2012.
- [8] N. Parikh, S. Boyd, *et al.*, “Proximal algorithms,” *Foundations and Trends in Optimization*, vol. 1, no. 3, 2014.
- [9] R. Bellman, “On the approximation of curves by line segments using dynamic programming,” *Communications of the ACM*, vol. 4, no. 6, 1961.
- [10] R. Bellman and R. Roth, “Curve fitting by segmented straight lines,” *Journal of the American Statistical Association*, vol. 64, no. 327, 1969.
- [11] R. Johansson, *System modeling & identification*. Prentice-Hall, Englewood Cliffs, NJ, 1993.
- [12] P. C. Hansen, “Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems,” *Numerical algorithms*, vol. 6, no. 1, 1994.
- [13] S. Levine and V. Koltun, “Guided policy search,” in *Int. Conf. Machine Learning (ICML), Atlanta*, 2013.
- [14] S. Levine, N. Wagener, and P. Abbeel, “Learning contact-rich manipulation skills with guided policy search,” in *Robotics and Automation (ICRA), IEEE Int. Conf.* IEEE, 2015.
- [15] D. A. Bristow, M. Tharayil, and A. G. Alleyne, “A survey of iterative learning control,” *IEEE Control Systems*, vol. 26, no. 3, 2006.

⁴The Kalman-filter implementation is often several orders of magnitude faster than solving the optimization problems with an iterative solver.