Discrete Inverse Problem Approach to Path Tracking in State Space Form

Matthew Harker and Gerhard Rath
Institute for Automation
University of Leoben
Leoben, Austria
matthew.harker@unileoben.ac.at

***Abstract*—This paper presents a new method for determining the ideal control signal to a state-space system such that one or more of its states track specified paths. The method is based on the concept of an inverse problem, whereby we determine the ideal input which is required to track the specified output. One of the key observations is that regularization must be applied to the solution of the inverse problem to obtain a unique solution, e.g., second order regularization is used to impose smoothness on the solution for the control. The method is verified on a synthetic example of a second order system whose position is specified and tracked according to the proposed algorithm. The results can be combined with a simple controller for path tracking as a form of model predictive control for real-world systems.**

***Keywords*—Path tracking; Inverse problem; Regularization**

I. Introduction

An important problem in mechanical engineering is the control of machines in order to follow given trajectories. Path following is a current problem, as it is an integral aspect of autonomous (or self-driving) vehicles [6], [7]. For example milling machines, tunnel cutting machines, or industrial robots have to follow a pre-defined path coordinating two or more axes. A frequent technical solution for this problem is to use state space control, or even each axis controlled separately, and to reduce the remaining tracking error with speed feed-forward [3]. Feed-forward compensation is simple to understand, does not require much additional computational effort, and provides remarkable improvement in the results. A more modern approach is model predictive control (MPC), where a recursive model of the dynamic system is combined with prediction of the control input [9]. With the solution presented in this paper, we follow the feed-forward approach. This allows us to keep the actual, preferably simple control scheme that can be processed with standard programmable logic control (PLC [2]). The contributions of this paper are:

1) A new method for the numerical solution of the inverse problem of obtaining an ideal control which induces a desired path of the system.
2) The new method is a least-squares approach which allows for sub-optimal solutions, i.e., solutions which track the desired path in a best fit sense, while the input is subject to regularization.
3) Verification of the computed control law by solving the direct problem with the control signal obtained from the solution to the inverse problem.

The method is verified with a second order system where the position of the system is specified by a piecewise continuous function. Results show that the path is tracked to a high accuracy, whereby the identified control signal remains reasonably smooth.

II. Preliminaries

A. Path Tracking as an Inverse Problem

Given the state-space form of a system of differential equations [3],

$$\dot{\boldsymbol{x}} = \mathsf{A}\boldsymbol{x} + \mathsf{B}\boldsymbol{u}, \tag{1}$$

the problem of path tracking can be formulated as obtaining a set of controls, $\boldsymbol{u}$, given the desired output of the system, $\boldsymbol{x}$. This can also be expressed as the inverse problem of obtaining a necessary input to the system which will produce a given output. To see the difficulty of this problem, we may rearrange the system as,

$$\mathsf{B}\boldsymbol{u} = \dot{\boldsymbol{x}} - \mathsf{A}\boldsymbol{x}. \tag{2}$$

In this form, we see immediately the difficulties in solving it. If we had the same number of controls as states, we may consider inverting the matrix B to obtain a solution. However, generally this is not the case, and further, it would also depend on all of the states, $\boldsymbol{x}(t)$ being fully specified. Ideally, we would like to specify only one state, e.g., $x_1(t) \approx \xi(t)$. It is however, of note, that when we have fewer controls than states, then we can write the control law as,

$$\boldsymbol{u} = \mathsf{B}^+ \left(\dot{\boldsymbol{x}} - \mathsf{A}\boldsymbol{x}\right) + \tilde{\mathsf{V}}\boldsymbol{\beta}(t), \tag{3}$$

where B^+ is the Moore-Penrose pseudo-inverse of B, the matrix $\tilde{\mathsf{V}}$ is a vector basis for the null-space of B, and $\boldsymbol{\beta}(t)$ is a set of arbitrary functions of the independent variable t. This shows that even if all states of $\boldsymbol{x}$ were fully specified, we would still have an infinite family of solutions for the controls. Thus, the path tracking problem can be more precisely formulated as: From the specified state, $x_1(t) \approx \xi(t)$, determine the remaining states of $\boldsymbol{x}(t)$ and a unique control from the space of all possible controls.

ISBN 978-80-261-0722-4, © University of West Bohemia, 2018

B. Discretization of the State Space Form

In order to discretize the state-space system in Equation (1), we first note that any state can be discretized directly as a vector,

$$\boldsymbol{x}_k = \begin{bmatrix} x_k(t_0) \\ x_k(t_1) \\ \vdots \\ x_k(t_f) \end{bmatrix} \tag{4}$$

Further, its derivative can be discretized as the matrix operation,

$$\dot{\boldsymbol{x}}_k \approx \mathsf{D}\boldsymbol{x}_k, \tag{5}$$

where the differentiation matrix is composed of numerical differentiation rules such as,

$$\mathsf{D} = \frac{1}{2h}\begin{bmatrix} -3 & 4 & -1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 1 & -4 & 3 \end{bmatrix}, \tag{6}$$

where h is the even spacing $h = t_k - t_{k-1}$. With these definitions, we may discretize the state vector with p states as,

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_p \end{bmatrix} \tag{7}$$

and in this manner the system of differential equations is discretized as,

$$(\mathsf{I}_p \otimes \mathsf{D})\,\boldsymbol{x} = (\mathsf{A} \otimes \mathsf{I}_n)\,\boldsymbol{x} + (\mathsf{B} \otimes \mathsf{I}_n)\,\boldsymbol{u}, \tag{8}$$

where $\otimes$ is the Kronecker product. For details see [8].

III. Variational Approach to Path Tracking

In order to formulate a variational approach to the path tracking problem, we must first establish what criteria to use for minimization. Firstly, the cost function for path tracking can be formulated as,

$$\int_{t_0}^{t_f} (x_1(t) - \xi(t))^2 \, \mathrm{d}t, \tag{9}$$

which quantifies the least-squares difference between the state $x_1(t)$ and the desired path $\xi(t)$. For convenience, this can be written in terms of the state vector as,

$$\int_{t_0}^{t_f} (\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}(t) - \xi(t))^2 \, \mathrm{d}t, \tag{10}$$

where $\boldsymbol{e}_1$ is a coordinate vector. Note also that any particular state could be specified, and not specifically $x_1(t)$. Further, in order to obtain a unique solution for the control [5], it is necessary to introduce the regularization term (or "penalty term"),

$$\mu^2 \int_{t_0}^{t_f} \left(u^{(j)}(t)\right)^2 \, \mathrm{d}t, \tag{11}$$

where μ is the regularization parameter. Clearly, if $j = 0$ and $\mu = 1$, this would be the cost function for optimal control, i.e., minimization of the magnitude of the control over the prescribed time interval. For the purposes of path tracking, it may be of use to invoke the case $j = 2$, which would penalize the magnitude of the second derivative of the control, and therefore induce smoothness in the control. But ideally, in the following, in the limit as $\mu \to 0$, we should obtain the same unique solution to the path following problem independent of the regularization order j.
With this formulation of the cost function and regularization, we can thereby formulate the path tracking problem with a single control as the variational problem of minimizing the functional,

$$\begin{aligned} J(t, \boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), u(t), \boldsymbol{\lambda}(t)) = \\ \frac{1}{2}\int_{t_0}^{t_f} (\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}(t) - \xi(t))^2 \, \mathrm{d}t + \frac{\mu^2}{2}\int_{t_0}^{t_f} (u(t))^2 \, \mathrm{d}t \\ - \int_{t_0}^{t_f} \boldsymbol{\lambda}^{\mathrm{T}}(t)\,(\dot{\boldsymbol{x}}(t) - \mathsf{A}\boldsymbol{x}(t) - \mathsf{B}u(t)) \, \mathrm{d}t \end{aligned} \tag{12}$$

The corresponding Euler-Lagrange equations with respect to $\boldsymbol{x}(t)$, $u(t)$ and $\boldsymbol{\lambda}(t)$ are respectively,

$$\begin{aligned} \boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}}\boldsymbol{x}(t) + \boldsymbol{e}_1\xi(t) + \mathsf{A}^{\mathrm{T}}\boldsymbol{\lambda}(t) + \dot{\boldsymbol{\lambda}}(t) = \boldsymbol{0} \\ \mu^2 u(t) + \mathsf{B}^{\mathrm{T}}\boldsymbol{\lambda}(t) = 0 \\ -\dot{\boldsymbol{x}}(t) + \mathsf{A}\boldsymbol{x}(t) + \mathsf{B}u(t) = \boldsymbol{0}. \end{aligned} \tag{13}$$

At this point, we can see the importance of the regularization term: If μ were zero, we would now have no way of uniquely determining $\boldsymbol{u}(t)$. Finally, if we assume that $\mu > 0$, we can (temporarily) eliminate $u(t)$, and write the Euler-Lagrange equations in the form,

$$\begin{bmatrix} \dot{\boldsymbol{x}}(t) \\ \dot{\boldsymbol{\lambda}}(t) \end{bmatrix} = \begin{bmatrix} \mathsf{A} & -\frac{1}{\mu^2}\mathsf{B}\mathsf{B}^{\mathrm{T}} \\ -\boldsymbol{e}_1\boldsymbol{e}_1^{\mathrm{T}} & -\mathsf{A}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(t) \\ \boldsymbol{\lambda}(t) \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{e}_1 \end{bmatrix} \xi(t) \tag{14}$$

This is a system of $2p$ linear differential equations. A total of $2p$ initial or boundary values should be specified to solve this set of equations. However, in this case, we only have specific information about $x_1(t)$, that is, it should be fully specified. To solve this equation, we may enforce that the boundary conditions on $x_1(t)$ match those of the specified function $\xi(t)$, i.e.,

$$\begin{array}{cc} x_1(t_0) = \xi(t_0) & x_1(t_f) = \xi(t_f) \\ x_1^{(1)}(t_0) = \xi^{(1)}(t_0) & x_1^{(1)}(t_f) = \xi^{(1)}(t_f) \\ \vdots & \vdots \\ x_1^{(p-1)}(t_0) = \xi^{(p-1)}(t_0) & x_1^{(p-1)}(t_f) = \xi^{(p-1)}(t_f) \end{array} \tag{15}$$

This provides sufficient information to solve the system. In the following, we derive a numerical solution to the path tracking problem.

IV. Numerical Method for Path Tracking

The matrix based approach to the discretization of the state-space form provides an efficient and effective means of solving the path tracking problem numerically. In discretized form, with evenly spaced abscissae, we have,

$$\int_{t_0}^{t_f} (x_1(t) - \xi(t))^2 \, \mathrm{d}t \sim \|\boldsymbol{x}_1 - \boldsymbol{\xi}\|_2^2 , \tag{16}$$

and for the control variable, we have,

$$\mu^2 \int_{t_0}^{t_f} \left(u^{(j)}(t)\right)^2 \, \mathrm{d}t \sim \mu^2 \left\|\mathsf{D}^{(j)}\boldsymbol{u}\right\|_2^2 , \tag{17}$$

where $\mathsf{D}^{(j)}$ is a matrix that computes a numerical approximation to the j^{th} derivative of $u(t)$. Finally, we extract a discretized state from the full state vector by a simple permutation matrix, whereby in our case,

$$\boldsymbol{x}_1 = \mathsf{P}_1\boldsymbol{x} = \begin{bmatrix}\mathsf{I}_n & 0 & \cdots & 0\end{bmatrix} \begin{bmatrix}\boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_p\end{bmatrix} \tag{18}$$

The solution is constrained to satisfy the discretized system of differential equations in Equation (8), i.e.,

$$(\mathsf{I}_p \otimes \mathsf{D})\, \boldsymbol{x} = (\mathsf{A} \otimes \mathsf{I}_n)\, \boldsymbol{x} + (\mathsf{B} \otimes \mathsf{I}_n)\, \boldsymbol{u}, \tag{19}$$

whereby the solution vector $\boldsymbol{x}$ also satisfies the boundary conditions of the form,

$$\mathsf{C}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{d}. \tag{20}$$

In this case, the matrix C and vector $\boldsymbol{d}$ define the boundary conditions in Equations (15). The discretization of the boundary conditions proceeds by denoting the matrix of the j^{th} derivative by the row partitioning,

$$\mathsf{D}^{(j)} = \begin{bmatrix}\left(\boldsymbol{d}_1^{(j)}\right)^{\mathrm{T}} \\ \vdots \\ \left(\boldsymbol{d}_n^{(j)}\right)^{\mathrm{T}}\end{bmatrix}. \tag{21}$$

Then the matrix C takes the form,

$$\mathsf{C} = \begin{bmatrix}\boldsymbol{e}_1 & \boldsymbol{e}_n & \boldsymbol{d}_1^{(1)} & \boldsymbol{d}_n^{(1)} & \cdots & \boldsymbol{d}_1^{(p-1)} & \boldsymbol{d}_n^{(p-1)}\end{bmatrix} \tag{22}$$

and the vector $\boldsymbol{d}$ takes the form,

$$\boldsymbol{d} = \begin{bmatrix}\xi(t_0) \\ \xi(t_f) \\ \xi^{(1)}(t_0) \\ \xi^{(1)}(t_f) \\ \vdots \\ \xi^{(p-1)}(t_0) \\ \xi^{(p-1)}(t_f)\end{bmatrix} \tag{23}$$

Firstly, in order to minimize the cost function and regularization terms, we require an explicit expression for $\boldsymbol{x}$. This is obtained by solving the discretized state-space form subject to the boundary conditions. That is, if we define the matrices,

$$\begin{aligned}\mathsf{L} &= (\mathsf{I}_p \otimes \mathsf{D}) - (\mathsf{A} \otimes \mathsf{I}_n) \\ \mathsf{M} &= (\mathsf{B} \otimes \mathsf{I}_n)\,,\end{aligned} \tag{24}$$

then the system of equations in (8) reads,

$$\mathsf{L}\boldsymbol{x} = \mathsf{M}\boldsymbol{u}. \tag{25}$$

This problem is resolved by computing the QR decomposition of the matrix C as,

$$\mathsf{C} = \mathsf{QR} = \begin{bmatrix}\hat{\mathsf{Q}} & \tilde{\mathsf{Q}}\end{bmatrix} \begin{bmatrix}\hat{\mathsf{R}} \\ 0\end{bmatrix} \tag{26}$$

where the matrices are partitioned such that $\hat{\mathsf{R}}$ is a full rank $2p \times 2p$ matrix [4]. We can then apply an orthogonal change of variables to the solution such that,

$$\boldsymbol{x} = \begin{bmatrix}\hat{\mathsf{Q}} & \tilde{\mathsf{Q}}\end{bmatrix} \begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{z}\end{bmatrix} \tag{27}$$

The boundary conditions then read,

$$\mathsf{R}^{\mathrm{T}}\mathsf{Q}^{\mathrm{T}}\boldsymbol{x} = \boldsymbol{d}. \tag{28}$$

whereby, with the substitution we have

$$\begin{aligned}\begin{bmatrix}\hat{\mathsf{R}}^{\mathrm{T}} & 0\end{bmatrix} \begin{bmatrix}\boldsymbol{y} \\ \boldsymbol{z}\end{bmatrix} &= \boldsymbol{d} \\ \hat{\mathsf{R}}^{\mathrm{T}}\boldsymbol{y} &= \boldsymbol{d}.\end{aligned} \tag{29}$$

Since $\hat{\mathsf{R}}$ is invertible, we have,

$$\boldsymbol{y} = \hat{\mathsf{R}}^{-\mathrm{T}}\boldsymbol{d}. \tag{30}$$

Thus, the constrained vector $\boldsymbol{x}$ takes the form,

$$\boldsymbol{x} = \hat{\mathsf{Q}}\hat{\mathsf{R}}^{-\mathrm{T}}\boldsymbol{d} + \tilde{\mathsf{Q}}\boldsymbol{z} \tag{31}$$

This describes the set of admissible functions which satisfy the given boundary conditions. We substitute this into the expression for the system of differential equations to obtain,

$$\mathsf{L}\left(\hat{\mathsf{Q}}\hat{\mathsf{R}}^{-\mathrm{T}}\boldsymbol{d} + \tilde{\mathsf{Q}}\boldsymbol{z}\right) = \mathsf{M}\boldsymbol{u} \tag{32}$$

We can solve this equation for $\boldsymbol{z}$ by means of the Moore-Penrose pseudo-inverse as,

$$\boldsymbol{z} = \left(\mathsf{L}\tilde{\mathsf{Q}}\right)^{+}\left(\mathsf{M}\boldsymbol{u} - \mathsf{L}\hat{\mathsf{Q}}\hat{\mathsf{R}}^{-\mathrm{T}}\boldsymbol{d}\right) \tag{33}$$

Finally, by substituting this expression into the expression for $\boldsymbol{x}$ we obtain the solution,

$$\boldsymbol{x} = \tilde{\mathsf{Q}}\left(\mathsf{L}\tilde{\mathsf{Q}}\right)^{+}\mathsf{M}\boldsymbol{u} + \left(\mathsf{I}_{pn} - \tilde{\mathsf{Q}}\left(\mathsf{L}\tilde{\mathsf{Q}}\right)^{+}\mathsf{L}\right)\hat{\mathsf{Q}}\hat{\mathsf{R}}^{-\mathrm{T}}\boldsymbol{d}. \tag{34}$$

This expression is the explicit solution for the system of differential equations, subject to the boundary conditions, dependent on the unknown input vector $\boldsymbol{u}$. For compactness, we may denote this as,

$$\boldsymbol{x} = \mathsf{G}\boldsymbol{u} + \boldsymbol{x}_0. \tag{35}$$

Finally, we substitute this expression into the cost function, that is, the sum of the terms in Equations (16) and (17), to obtain,

$$\epsilon(\boldsymbol{u}) = \|\mathsf{P}_1\left(\mathsf{G}\boldsymbol{u} + \boldsymbol{x}_0\right) - \boldsymbol{\xi}\|_2^2 + \mu^2\left\|\mathsf{D}^{(j)}\boldsymbol{u}\right\|_2^2 \tag{36}$$

Since this is the sum of two squared norms, the vectors can be stacked together, and the cost function can be rearranged to read,

$$\epsilon(\boldsymbol{u}) = \left\|\begin{bmatrix}\mathsf{P}_1\mathsf{G} \\ \mu\mathsf{D}^{(j)}\end{bmatrix}\boldsymbol{u} - \begin{bmatrix}\boldsymbol{\xi} - \mathsf{P}_1\boldsymbol{x}_0 \\ \boldsymbol{0}\end{bmatrix}\right\|_2^2. \tag{37}$$

For a fixed value of μ, this is now a standard least-squares problem for the unknown control, $\boldsymbol{u}$, and can be solved directly using any appropriate least squares solver [1].

V. Verification

In order to verify the proposed algorithm we investigated the second order system,

$$m\ddot{x}(t) + b\dot{x}(t) + cx(t) = u(t) \tag{38}$$

for which we have the system matrices,

$$\mathsf{A} = \begin{bmatrix} 0 & 1 \\ -\frac{c}{m} & -\frac{b}{m} \end{bmatrix} \qquad \text{and} \qquad \mathsf{B} = \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}. \tag{39}$$

For the numerical test, the values,

$$m = 1, \qquad c = 8, \qquad \text{and} \qquad b = 0.04, \tag{40}$$

were used. To test the path tracking, the function to be followed was specified as the piecewise function,

$$\xi(t) = \begin{cases} \frac{1}{2}(1 - \cos\left(\frac{\pi}{2}t\right)) & \text{for} \quad t \in [0, 2) \\ \frac{3}{4} + \frac{1}{4}\cos(\pi t) & \text{for} \quad t \in [2, 3) \\ \frac{1}{4}(1 - \cos(\pi t)) & \text{for} \quad t \in [3, 4] \end{cases} \tag{41}$$

The physical interpretation of the problem is depicted in Figure 1. The problem was solved with a second order derivative in the regularization term, thus providing regularization in the form of smoothing of the control. The results of the proposed algorithm are shown in Figure 2. The algorithm delivers a solution for the position which tracks the desired path to very high accuracy. The accuracy of the actual path can be decreased by increasing the regularization parameter, μ. That is, the larger μ is the more weight is given to smoothing out the control variable, and the less accurately the path is tracked. However, ideally, we are interested in the limit solution $\mu \rightarrow 0$. In Figure 3, the solution obtained for the control required to follow the specified path is shown.

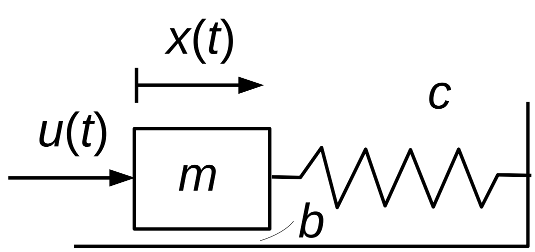

Fig. 1. The path to be followed of the second order system, the solution for the path following problem, and the velocity of the mass due to the control.

VI. Conclusions and Future Work

This paper presented a new inverse problem approach to the problem of path tracking, whereby regularization was used to obtain a unique solution to the problem. The result of the algorithm can be interpreted as a combination of the path data and the predicted dynamical behaviour that is calculated in advance and can be used in a feed-forward scheme. Underlying controllers then have the task to correct for model errors or unpredictable load conditions. Future work includes implementing further constraints such that control limitations can be considered.

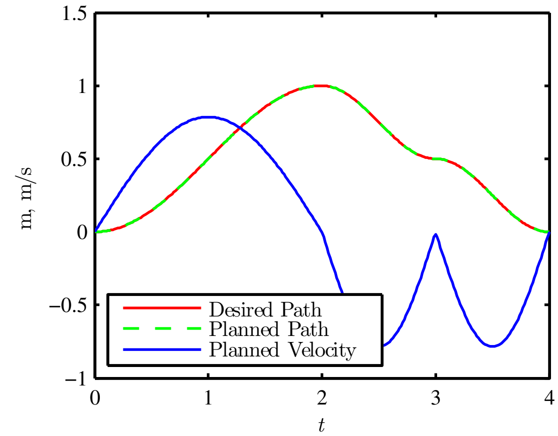

Fig. 2. The path to be followed of the second order system, the solution for the path following problem, and the velocity of the mass due to the control.

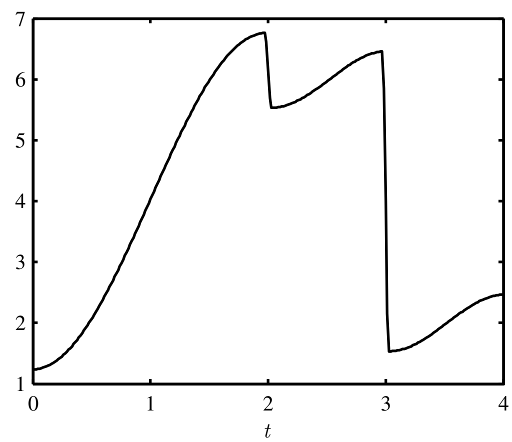

Fig. 3. The solution of the inverse problem, i.e., the control required to obtain the desired output.

Acknowledgment

This project was partially funded by the *EIT RawMaterials* project *Rock Vader*.

References

[1] Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996.

[2] W. Bolton. *Programmable Logic Controllers*. Newnes, 5 edition, 2009.

[3] G.F. Franklin, J.D. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Addison-Wesley, third edition, 1998.

[4] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, 4^{th} edition, 2013.

[5] R.E. Kalman. *The Theory of Optimal Control and the Calculus of Variations*, chapter 16. University of California Press, 1963.

[6] N.R. Kapania and J.C. Gerdes. Design of a feedback-feedforward steering controller for accurate path tracking and stability at the limits of handling. *Vehicle System Dynamics*, 53(12):1687–1704, 2015.

[7] H. Mouria, S. Satoha, H. Furushoa, and M. Nagai. Investigation of automatic path tracking using an extended Kalman filter. *JSAE Review*, 23:61–67, 2002.

[8] G. Rath and M. Harker. Global least squares solution for LTI system response. In *22nd International Conference on Applied Electronics*, 2017.

[9] J.B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, (June 2000):38–52, 2000.