Differential Dynamic Programming for Optimal Estimation

Marin Kobilarov[1], Duy-Nguyen Ta[2], Frank Dellaert[3]

***Abstract*— This paper studies an optimization-based approach for solving optimal estimation and optimal control problems through a unified computational formulation. The goal is to perform trajectory estimation over extended past horizons and model-predictive control over future horizons by enforcing the same dynamics, control, and sensing constraints in both problems, and thus solving both problems with identical computational tools. Through such systematic estimation-control formulation we aim to improve the performance of autonomous systems such as agile robotic vehicles. This work focuses on sequential sweep trajectory optimization methods, and more specifically extends the method known as differential dynamic programming to the parameter-dependent setting in order to enable the solutions to general estimation and control problems.**

I. Introduction

The standard practice in the control of autonomous vehicles is to treat *perception* and *control* as separate problems, often solved using vastly different computational techniques. In order to cope with analytical and computational complexity a common approach is to simplify or ignore dynamical and control constraints during estimation, and similarly to not fully capture sensing constraints during control. From a theoretical point of view, we argue that this approach is severely limited as it does not reflect the inherent duality between estimation and control. From a practical point of view, one should expect an increase in performance if the full physical dynamics and constraints were systematically enforced. This could be especially relevant as speed and agility are becoming increasingly important in almost any mode of locomotion–in air, on wheels, legs, or under water.

This paper aims to develop a unified computational approach for both estimation and control problems through a single nonlinear optimization formulation subject to nonlinear differential constraints and control constraints. The formulation captures problems including system identification, environmental mapping over a past horizon, and model-predictive control over a future horizon. Estimation problems are thus regarded as trajectory *smoothing* [7] while control problems as *model-predictive-control* (MPC) [28], [20], [18]. Our particular focus is on *differential dynamic programming* (DDP) [21] which is one of the most effective *sweep* optimal control methods [4], i.e. methods that optimize in a backward-forward sequential fashion in order to exploit the additive and recursive problem structure. While DDP is a standard approach for control, we show that it can be naturally extended to optimal estimation as well. A *parameter-dependent differential dynamic programming* (PDDP) approach is thus proposed to solve simultaneous trajectory and parameter optimization. We then show that the computational techniques for ensuring convergence in the control setting carry over to optimal estimation problems.

This work explores the use of detailed second-order dynamical models (typically employed for control) for estimation as opposed to the standard practice in robotics based on kinematic first-order models with velocity inputs. Early methods for Simultaneous Localization and Mapping (SLAM) [14] include landmark positions in an extended Kalman filter (EKF) formulation but often exhibit inconsistent estimates due to linearization over time [37], [19], [17]. Unlike filtering, *Smoothing and mapping* (SAM) optimizes over an extended trajectory and alleviates such inconsistencies. Many modern approaches leverage various graph inference techniques. SAM, for example, has been shown to be equivalent to inference on graphical models known as *factor graphs* [12], [13]. Recent estimation methods based on related ideas have focused on fast performance for real-time navigation [10], [25], [24], [23], [31], large-scale problems with massive amount of sensor data [1], [6], [33], [34], [22], [11], and long-term navigation [35], [36], [8], [9], [32].

However, a unified *computational approach* for performing both nonlinear smoothing for perception and nonlinear model-predictive-control is rarely used. This is despite that both problems stem from a closely related principle of optimality [7], [38], [39] which in the linear-Gaussian case is the well-known duality between the celebrated linear quadratic regulator (LQR) and linear Gaussian estimator (LGE).

Assumptions.: In this work we only consider uncertainty during estimation, and employ the optimally estimated model (including internal parameters and environmental structure) for deterministic optimal control. Furthermore, environmental estimation is performed assuming perfect data association, i.e. that features (such as landmarks) have been successfully processed and associated to prior data.

II. Problem Formulation

Our focus is on systems with nonlinear dynamics described by a *dynamic state* $x(t)\in X$, *control inputs* $u(t)\in U$, *process uncertainty* $w(t)\in\mathbb{R}^\ell$, and *static parameters* $\rho\in\mathbb{R}^m$ related to the internal system structure (i.e. unknown/uncertain kinematic, dynamic, and calibration terms) or external environment structure (i.e. landmark or obstacle locations)

$$\rho=(\rho_{\text{int}},\rho_{\text{ext}}),\begin{cases}\rho_{\text{int}} \text{ – internal parameters,}\\ \rho_{\text{ext}} \text{ – external parameters.}\end{cases}$$

[1]Marin Kobilarov is with Faculty of Mechanical Engineering, Johns Hopkins University, Baltimore, MD, USA marin@jhu.edu

[2]Duy-Nguyen Ta and [3]Frank Dellaert are with the Center for Robotics and Intelligent Machines, Georgia Institute of Technology, USA duynguyen, dellaert@gatech.edu

The time-horizon of interest is denoted by $[t_0, t_f]$ during which the systems obtains N measurements, with measurement at time t_i denoted by z_i. The complete process is defined by

$$\dot{x}(t) = f\left(x(t), u(t), w(t), \rho, t\right), \quad \text{(dynamics)} \quad (1)$$

$$z_i = h_i(x(t_i), \rho) + v_i, \quad i = 1, \ldots, N \quad \text{(measurements)} \quad (2)$$

$$u(t) \in U, \quad x(t) \in X, \quad \text{(constraints)} \quad (3)$$

where f and h_i are nonlinear functions, and $v_i \in Z$ is a noise term.

We are interested in solving control and estimation problems involving the minimization of the general cost:

$$J(x(\cdot), \xi(\cdot), \rho) = \varphi(x(t_f), \rho) + \int_{t_0}^{t_f} L(t, x(t), \xi(t), \rho) dt, \quad (4)$$

where the variables $\xi(t)$ can either denote the controls, i.e. $\xi(t) \triangleq u(t)$, for control problems, or $\xi(t) \triangleq w(t)$ for estimation problems, as specified next.

State and parameter estimation.: When solving estimation problems, the cost $J(x(\cdot), \xi(\cdot), \rho) \triangleq J_{\text{est}}(x(\cdot), w(\cdot), \rho)$ is defined as the negative log-likelihood

$$J_{\text{est}} = -\log\Big\{p(x(t_0), \rho) \cdot \prod_{i=1}^{N} p\left(x(t_i) | x(t_{i-1}), u_{[t_{i-1}, t_i]}, \rho\right) \cdot p(z_i | x(t_i), \rho)\Big\}, \quad (5)$$

where $p(x_0, \rho)$ is a known prior probability density on the initial state x_0 and parameters ρ, where $p(x_i | x_{i-1}, u_{[t_{i-1}, t_i]}, \rho)$ is a state transition density for moving from state x_{i-1} to state x_i after applying control signal $u([t_{i-1}, t_i])$ with parameters ρ, and where $p(z|x, \rho)$ is the likelihood of obtaining a measurement z from state x with parameters ρ. Minimizing J_{est} corresponds to a trajectory *smoothing* problem with unknowns $[x(t), w(t), \rho]$. We assume that the controls $u(t)$ applied to the system are known.

Model-predictive control (MPC).: For MPC problems the cost $J(x(\cdot), \xi(\cdot), \rho) \triangleq J_{\text{mpc}}(x(\cdot), u(\cdot))$ is often defined as

$$J_{\text{mpc}} = \frac{1}{2}\|x(t_f) - x_f\|^2_{Q_f} + \int_{t_0}^{t_f}\left[q(t, x(t)) + \frac{1}{2}\|u(t)\|^2_{R(t)}\right] dt, \quad (6)$$

where $q(t, x)$ is a given nonlinear function, while the matrices Q_f and $R(t)$ provide physically meaningful cost weights. This is a *prediction* problem with unknowns $[x(t), u(t)]$. Here we have assumed nominal process uncertainty $w(t)$, i.e. $w(t) = 0$.

Active sensing.: During active sensing the vehicle computes a model-predictive future trajectory by balancing control effort and likelihood of estimated state and parameters. This is accomplished by setting $J(x(\cdot), \xi(\cdot), \rho) \triangleq J_{\text{as}}(x(\cdot), u(\cdot), \rho)$ where

$$J_{\text{as}} = \mathbb{E}_{w(\cdot), v_{1:N}}\left[J_{\text{est}}(x(\cdot), w(\cdot), \rho) + \beta J_{\text{mpc}}(x(\cdot), u(\cdot))\right], \quad (7)$$

for some $\beta > 0$ or, practically speaking, as the expected combined estimation-control cost over all possible future states and measurements. Here, $w(\cdot)$ denotes a continuous process noise signal over $[t_0, t_f]$, while $v_{1:N} \triangleq \{v_1, \ldots, v_N\}$ denotes a finite sequence of measurement noise terms. For nonlinear systems, the expectation (7) cannot be computed in closed form. One solution is to use the approximate cost

$$\widehat{J}_{\text{as}} = \sum_{j=1}^{P} c_j \left[J_{\text{est}}(x^j(\cdot), w^j(\cdot), \rho) + \beta J_{\text{mpc}}(x^j(\cdot), u(\cdot))\right],$$

based on P samples $(w^j(\cdot), v^j_{1:N})$ for $j = 1, \ldots, P$, with weights c_j such that $\sum_{j=1}^{P} c_j = 1$. Here, J_{est} is computed using measurements $z^j_i = h(x^j_i, \rho) + v^j_i$ and the sampled trajectories $x^j(\cdot)$ satisfy $\dot{x}^j(t) = f(t, x^j(t), u(t), w^j(t), \rho)$. The samples can either be chosen independently and identically distributed in which case we have $c_j = 1/P$ or, for instance, using an unscented transform.

Adaptive model-predictive control.

Our proposed strategy is to solve both an optimal estimation and an optimal control problem at each sampling time t, by first performing estimation over a past horizon of T_{est} seconds and then applying MPC over a future horizon of T_{mpc} seconds. If t denotes the current time, then first J_{est} is optimized over the interval $[t - T_{\text{est}}, t]$ to update the current state $x(t)$ and parameters ρ, after which J_{mpc} is optimized over $[t, t + T_{\text{mpc}}]$ to compute and apply the optimal controls $u(t)$. Next, we specify a common numerical approach for optimizing either (5) or (6). Although our general methodology also applies to active sensing costs (7), this paper develops examples only for costs (5) and (6).

The finite-dimensional optimization problem.

The proposed numerical methods are based on *discrete trajectories* $x_{0:N} \triangleq \{x_0, x_1, \ldots, x_N\}$ and $\xi_{0:N-1} \triangleq \{\xi_0, \xi_1, \ldots, \xi_{N-1}\}$ using time discretization $\{t_0, t_1, \ldots, t_N\}$ with $t_N = t_f$ where $x_i \approx x(t_i)$ and $\xi_i \approx \xi(t_i)$. With these definitions, optimizing the functional (4) will be accomplished using the finite-dimensional minimization of the objective

$$J_N(x_{0:N}, \xi_{0:N-1}, \rho) \triangleq L_N(x_N, \rho) + \sum_{i=0}^{N-1} L_i(x_i, \xi_i, \rho), \quad (8)$$

$$\text{subject to:} \qquad x_{i+1} = f_i(x_i, \xi_i, \rho), \quad (9)$$

where $L_i \approx \int_{t_i}^{t_{i+1}} L dt$ encodes the cost along the i-th time stage, $L_N \equiv \varphi$ is the terminal cost, f_i encodes the update step of a numerical integrator from t_i to t_{i+1}. We again underscore that the term ξ has a different meaning based on whether J defines an estimation or a control problem. In particular, during estimation over a past-horizon the parameters ρ and uncertainties $w_{0:N-1}$ are treated as unknowns and hence $\xi_i \triangleq w_i$, while during optimal control over a future horizon the unknowns are the controls $u_{0:N-1}$, and hence $\xi_i \triangleq u_i$.

For computational convenience it is often assumed that the state and parameters have a Gaussian prior, i.e. $x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_{x_0})$ and $\rho \sim \mathcal{N}(\hat{\rho}, \Sigma_\rho)$ and that uncertainties are Gaussian, i.e. $w_i \sim \mathcal{N}(0, \Sigma_{w_i})$ and $v_i \sim \mathcal{N}(0, \Sigma_{v_i})$. The *esti-*

mation cost (5) is then equivalently expressed as

$$J_{\text{est}}=\frac{1}{2}\Big\{\|x_0-\hat{x}_0\|^2_{\Sigma^{-1}_{x_0}}+\|\rho-\hat{\rho}\|^2_{\Sigma^{-1}_{\rho}}+\sum_{i=0}^{N-1}\|w_i\|^2_{\Sigma^{-1}_{w_i}} + \sum_{i=1}^{N}\|z_i-h_i(x_i,\rho)\|^2_{\Sigma^{-1}_{v_i}}\Big\}. \tag{10}$$

III. Parameter-dependent Sweep Methods

Two standard types of numerical methods are applicable for solving the discrete optimal control problem (8)–(9): nonlinear programming using either collocation or multiple-shooting which optimizes directly over all variables, and stage-wise sweep methods which explicitly remove the trajectory constraint (9), e.g. by solving for the associated multipliers recursively.

Nonlinear programming [16], e.g. based on sequential-quadratic-programming (SQP) or interior-point (IP) methods are applicable as long as problem sparsity is exploited, i.e. by specifying Jacobian structure, to handle problems of reasonable size (e.g. a trajectory with $Nn > 100$).

Our focus will be on the latter type of methods. Instead of formulating an $[N(n+c)+m]$-dimensional monolithic program it is possible to explicitly factor out the trajectory constraints in a recursive manner and solve N smaller problems of dimension $[n+c]$ with m parameters and one m-dimensional program. From the optimal control point of view, *Stage-wise Newton method* (SN) [2] and *differential dynamic programming* (DDP) [21], [29] are the standard lines of attack in this context. From the estimation point of view, probably the most widely used approach to the smoothing problem is the discrete-time Rauch-Tung-Striebel (RTS) smoother [4], [7], [15], which is based on linearizing the dynamics and replacing nonlinear cost terms with their first-order Taylor series expansion. Keeping first-order terms only has been motivated by the least-squares form of the cost amenable to Gauss-Newton methods. Note that the Gauss-Newton approach exhibits quadratic convergence under the condition that either when the cost residual is small, or when the Hessian matrices have small eigenvalues.

While SN and DDP are common in control, they are equally capable of solving estimation problems and there is a reason to believe that they should outperform Gauss-Newton methods for highly non-linear problems by considering higher-order terms.

The *advantages* of sweep methods is that the dimensionality is directly reduced, that the true nonlinear dynamics is used in evolving the trajectory, that second-order information is exploited, and that stage-wise (localized) as opposed to a global regularization can be applied for finding a suitable search direction. The *disadvantage* is that complex state inequality constraints cannot be systematically enforced, something that active-set SQP and IP methods are specifically designed to handle [3]. In addition, when the constant parameter ρ includes a large number of landmarks it has been shown that explicitly factoring out the trajectory x can actually reduce efficiency by reducing sparsity and precluding optimally-ordered factorization methods. We thus expect that the proposed methods would be most effective for coupled estimation and control over a short horizons.

Linearization

The variational solution that we will develop will require infinitesimal relationship between state, control, and parameter variations in view of the dynamics. This can be written according to

$$\delta x_{i+1} = A_i \cdot \delta x_i + B_i \cdot \delta\xi_i + C_i \cdot \delta\rho. \tag{11}$$

and obtained as follows:

Explicit linearization: When the dynamics is provided in the explicit form $x_{i+1} = f_i(x_i, \xi_i, p)$ then the linearization is obtained by

$$A_i \equiv \partial_x f_i, \quad B_i \equiv \partial_\xi f_i, \quad C_i \equiv \partial_\rho f_i.$$

When analytical derivatives are not available, or when f_i is encoded by e.g. a complex black-box physics engine, these Jacobians are computed using finite differences.

Implicit Constraint Linearization: When the dynamics corresponds to an implicit analytical relationship

$$c_i(x_{i+1}, x_i, \xi_i, p) = 0,$$

we have $A_i = (D_1 c_i)^{-1}(D_2 c_i), B_i = (D_1 c_i)^{-1}(D_3 c_i), B_i = (D_1 c_i)^{-1}(D_4 c_i)$ assuming $D_1 c_i$ is full-rank.

Parameter-dependent Differential Dynamic Programming

The DDP approach is extended to the parameter-dependent in two steps: 1) the state x is augmented using the parameters ρ and the standard DDP-Backward sweep is applied to the augmented system; 2) parameter variations $\delta\rho$ are computed in a special manner and applied only once at the start of DDP-Forward while state variations δx_i are updated in the standard way.

Let the *augmented state* $\bar{x} \in X \times \mathbb{R}^c$ be defined as $\bar{x}_i = (x_i, \rho)$. The augmented discrete dynamics function $\bar{f}_i$ is then

$$\bar{f}_i(\bar{x}_i, \xi_i) = \begin{bmatrix} f_i(x_i, \xi_i, \rho) \\ \rho \end{bmatrix},$$

and its corresponding augmented state Jacobian $\bar{A}_i \triangleq \partial_{\bar{x}} \bar{f}_i$ is

$$\bar{A}_i = \begin{bmatrix} A_i & C_i \\ 0 & I \end{bmatrix}.$$

With these definitions we can apply DDP to the augmented system by first defining the *augmented cost-to-go* from state x_i at time-stage i using inputs $\xi_{i:N-1}$ by

$$J_i(\bar{x}_i, \xi_{i:N-1}) = L_N(x_N, \rho) + \sum_{k=i}^{N-1} L_k(x_k, \xi_k, \rho),$$

where $x_{i+1} = f_i(x_i, \xi_i, \rho)$. The *optimal value function* at time-stage i is denoted by $\mathcal{V}_i(\bar{x}_i)$ and defined according to

$$\mathcal{V}_i(\bar{x}_i) = \min_{\xi_{i:N-1}} J_i(\bar{x}_i, \xi_{i:N-1}).$$

The value function can be expressed recursively through the

HJB equations according to

$$\mathcal{V}_i(\bar{x}_i) = \min_{\xi} \left\{ \mathcal{V}_{i+1}\left(\bar{f}_i(\bar{x}_i, \xi)\right) + L_i(\bar{x}_i, u_i) \right\}. \tag{12}$$

Let $Q_i(\bar{x}, \xi)$ denote the unoptimized value function given by

$$Q_i(\bar{x}, \xi) = \mathcal{V}_{i+1}\left(\bar{f}_i(\bar{x}, u)\right) + L_i(\bar{x}, u).$$

In DDP one computes the controls u_i to minimize Q_i using a local second-order expansion with a resulting cost-change given by

$$\Delta Q_i \approx \frac{1}{2} \begin{bmatrix} 1 \\ \delta\bar{x}_i \\ \delta\xi_i \end{bmatrix} \begin{bmatrix} 0 & \nabla_{\bar{x}} Q_i^T & \nabla_{\xi} Q_i^T \\ \nabla_{\bar{x}} Q_i & \nabla_{\bar{x}}^2 Q_i & \nabla_{\bar{x}\xi} Q_i \\ \nabla_{\xi} Q_i & \nabla_{\xi\bar{x}} Q_i & \nabla_{\xi}^2 Q_i \end{bmatrix} \begin{bmatrix} 1 \\ \delta\bar{x}_i \\ \delta\xi_i \end{bmatrix}.$$

By the principle optimality, $\delta\xi_i$ should minimize ΔQ_i which results in the condition

$$\delta\xi_i^* = K_i \cdot \delta\bar{x}_i + \alpha_i k_i, \tag{13}$$

where $K_i = -\nabla_{\xi}^2 Q_i^{-1} \nabla_{\xi\bar{x}} Q_i, k_i = -\nabla_{\xi}^2 Q_i^{-1} \nabla_{\xi} Q_i$ for a chosen step-size $\alpha_i > 0$ [21]. The terms K_i and k_i are computed during the backward sweep and used to update the inputs ξ_i (using $\delta\xi_i$) which in turn are used to update that states x_i during the forward sweep:

PDDP-Backward

$\mathcal{V}_{\bar{x}} := \nabla_{\bar{x}} L_N, \quad \mathcal{V}_{\bar{x}\bar{x}} := \nabla_{\bar{x}}^2 L_N$

For $k = N-1 \to 0$

$Q_{\bar{x}} := \nabla_{\bar{x}} L_i + \bar{A}_i^T \mathcal{V}_{\bar{x}},$

$Q_{\xi} := \nabla_{\xi} L_i + B_i^T \mathcal{V}_x$

$Q_{\bar{x}\bar{x}} := \nabla_{\bar{x}}^2 L_i + \bar{A}_i^T (\mathcal{V}_{\bar{x}\bar{x}}) \bar{A}_i + \nabla_{\bar{x}}^2 f \cdot \mathcal{V}_{\bar{x}}$

$Q_{\xi\xi} := \nabla_{\xi}^2 L_i + B_i^T (\mathcal{V}_{\bar{x}\bar{x}}) B_i + \nabla_{\xi}^2 f \cdot \mathcal{V}_{\bar{x}}$

$Q_{\xi\bar{x}} := \nabla_{\xi\bar{x}} L_i + B_i^T \begin{bmatrix} \mathcal{V}_{xx} A_i & \mathcal{V}_{xx} C_i + \mathcal{V}_{x\rho} \end{bmatrix} + \nabla_{\xi\bar{x}} f \cdot \mathcal{V}_{\bar{x}}$

Choose $\mu_i > 0$ s.t. $\widetilde{Q}_{\xi\xi} \triangleq Q_{\xi\xi} + \mu_i I > 0$

$k_i = -\widetilde{Q}_{\xi\xi}^{-1} Q_{\xi}, \quad K_i = -\widetilde{Q}_{\xi\xi}^{-1} Q_{\xi\bar{x}}$

$\mathcal{V}_{\bar{x}} = Q_{\bar{x}} + {K_i}^T Q_{\xi}, \quad \mathcal{V}_{\bar{x}\bar{x}} = Q_{\bar{x}\bar{x}} + {K_i}^T Q_{\xi\bar{x}}$

The key difference between PDDP and DDP is that while in standard DDP one starts with setting $\delta x_0 = 0$ in the forward sweep, i.e. the initial state is known, in PDDP we actually set $\delta\bar{x}_0 = (0, \delta\rho^*)$ where $\delta\rho^*$ is a non-zero optimal change in the parameters.

This optimal change is computed through the optimization

$$\delta\rho^* = \min_{\delta\rho} \Delta\mathcal{V}_0(\bar{x}_0), \tag{14}$$

where

$$\Delta\mathcal{V}_i \triangleq \nabla_{\bar{x}} \mathcal{V}_i^T \delta\bar{x}_i + \frac{1}{2} \delta\bar{x}_i^T \nabla_{\bar{x}}^2 \mathcal{V}_i \, \delta\bar{x}_i,$$

with

$$\nabla_{\bar{x}} \mathcal{V} = \begin{bmatrix} \nabla_x \mathcal{V} \\ \nabla_{\rho} \mathcal{V} \end{bmatrix}, \quad \nabla_{\bar{x}}^2 \mathcal{V} = \begin{bmatrix} \nabla_x^2 \mathcal{V} & \nabla_{x\rho} \mathcal{V} \\ \nabla_{\rho x} \mathcal{V} & \nabla_{\rho}^2 \mathcal{V} \end{bmatrix}.$$

With these definition the solution to (14) reduces to

$$\delta\rho^* = -[\nabla_{\rho}^2 \mathcal{V}_0]^{-1} \nabla_{\rho} \mathcal{V}_0$$

since $\delta x_0 = 0$. In practice, regularization is employed to ensure convergence, when $\nabla_{\rho}^2 \mathcal{V}_0$ is not positive definite and invertible. The forward sweep is implemented using:

PDDP-Forward

Choose $\nu > 0$ s.t. $\widetilde{\mathcal{V}}_{\rho\rho} \triangleq \mathcal{V}_{\rho\rho} + \nu I > 0$

Cholesky-solve for $d \in \mathbb{R}^m$: $\widetilde{\mathcal{V}}_{\rho\rho} d = -\mathcal{V}_{\rho}$

Do: select step-size α

$\delta x_0 = 0, \quad \delta\rho = \alpha d, \quad \mathcal{V}_0' = 0$

$\rho' = \rho + \delta\rho$

For $i = 0 \to N-1$

$\xi_i' = \xi_i + \alpha k_i + K_i \begin{bmatrix} \delta x_i \\ \delta\rho \end{bmatrix}$

$x_{i+1}' = f_i(x_i', \xi_i', \rho')$

$\delta x_{i+1} = x_{i+1}' - x_{i+1}$

$\mathcal{V}_0' = \mathcal{V}_0' + L_i(x_i', \xi_i', \rho')$

$\mathcal{V}_0' = \mathcal{V}_0' + L_N(x_N', \rho')$

Until $(\mathcal{V}_0' - \mathcal{V}_0)$ is sufficiently negative

The step-size α is chosen using Armijo's rule [2] to ensure that the resulting controls $u_{0:N-1}'$ yield a sufficient decrease in the cost $\mathcal{V}_0' - \mathcal{V}_0$, where $\mathcal{V}_0$ is the computed value function in the previous iteration. In practice, second-order terms involving the dynamics (i.e. $\nabla_{\bar{x}}^2 f_i, \nabla_{\xi}^2 f_i, \nabla_{\xi\bar{x}} f_i$) can be ignored as long as they have small eigenvalues, or when $\mathcal{V}_{\bar{x}}$ is small. The complete algorithm consists of iteratively sweeping the trajectory backward and forward until convergence:

Optimize$(x_{0:N}, \xi_{0:N-1}, \rho)$

Iterate until convergence

PDDP-Backward

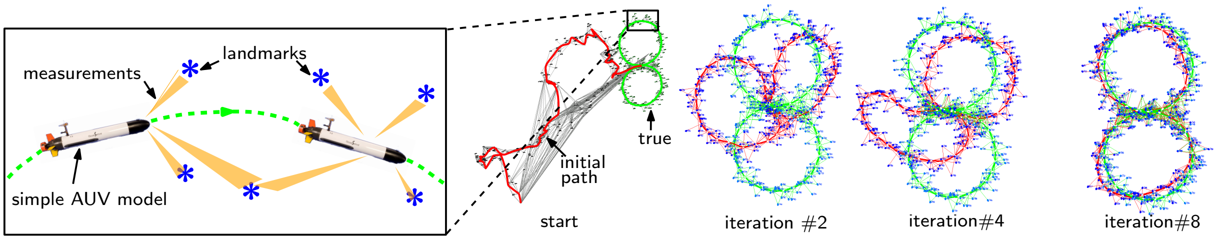
PDDP-Forward

Convergence: The algorithm is guaranteed to converge to a local minimum at which $\|\nabla_u Q_i\| \approx 0$ for all $i = 0, \ldots, N-1$ and $\|\nabla_{\rho} \mathcal{V}_0\| \approx 0$. This is ensured by employing regularization since the chosen steps δu_i and $\delta\rho$ result in reduction of the total cost following an argument employed in the original DDP algorithm [27]. In particular, the additional variation $\delta\rho$ results in the approximate change $\Delta_{\rho} \mathcal{V}_0 = -\alpha \mathcal{V}_{\rho}^T (\widetilde{\mathcal{V}}_{\rho\rho})^{-1} \mathcal{V}_{\rho} < 0$ for $\mathcal{V}_{\rho} \neq 0$.

IV. Optimization on the motion group SE(3)

In many practical applications involving vehicle models the n-dimensional state space can be decomposed as $X = SE(3) \times A$, where $SE(3)$ denotes the space of rigid-body motions and $A \subset \mathbb{R}^{n-6}$ is a vector space. Accordingly, the state is decomposed as $x = (g, a)$ where $g \in SE(3)$ and $a \in A$. In such cases, it is preferable to perform trajectory optimization directly on X rather than choosing coordinates such as Euler angles which have singularities.

Methods such as SQP, SN and DDP can be easily applied to optimization on the manifold X. This is accomplished using two modifications: using a *trivialized gradient* $d_g L \in$



a) AUV navigation mock-up: observing landmarks b) parameter-dependent DDP iterations computing optimal inputs and parameters

Fig. 1. Estimation and mapping using parameter-dependent DDP applied to a mock-up AUV navigation task.

$\mathbb{R}^6$ in place of the standard gradient $\nabla_g L$ of a given function $L: SE(3) \to \mathbb{R}$, and using *trivialized variations* $dg \in \mathbb{R}^6$ instead of the standard variation δg. This is necessary since both $\nabla_g L \in T_g^* SE(3)$ and $\delta g \in T_g SE(3)$ are matrices [1] (as opposed to vectors) and are not compatible with the vector Calculus employed by standard optimization methods. The trivialized gradient is defined by

$$d_g L \triangleq \nabla_V \Big|_{V=0} L(g\, \exp(V)), \tag{15}$$

for some $V \in \mathbb{R}^6$, where $\exp: \mathbb{R}^6 \to SE(3)$ is the standard exponential map [30]. The trivialized variation is defined by

$$dg \triangleq (g^{-1}\delta g)^\vee,$$

where the map $(\cdot)^\vee : \mathfrak{se}(3) \to \mathbb{R}^6$ is the inverse of the hat operator $\widehat{\cdot} : \mathbb{R}^6 \to \mathfrak{se}(3)$ which for a given $V = (\omega, \nu)$ is

$$\widehat{V} = \begin{bmatrix} \widehat{\omega} & \nu \\ 0_{1\times 3} & 0 \end{bmatrix}, \quad \widehat{\omega} = \begin{bmatrix} 0 & -w_3 & w_3 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{bmatrix}. \tag{16}$$

In essence, if g corresponds to rotation $R \in SO(3)$ and position $p \in \mathbb{R}^3$, the reduced variation is simply $dg = [(R^T \delta R)^\vee, \; R^T \delta p]^T$.

Applying DDP on X is then accomplished by defining A_i, B_i, C_i so that

$$dx_{i+1} = A_i dx_i + B_i \delta u_i + C_i \delta\rho, \tag{17}$$

where $dx_i \triangleq (dg_i, \delta a_i)$, employing $d_x L \triangleq (d_g L, \; \nabla_a L)$ instead of $\nabla_x L$ during the Backward sweep, and using

$$dx_{i+1} = (\log(g_{i+1}^{-1} g'_{i+1}), a'_{i+1} - a_{i+1})$$

instead of $\delta x_{i+1} = x'_{i+1} - x_{i+1}$ during the Forward sweep. Here the logarithm $\log : SE(3) \to \mathbb{R}^6$ is defined as the inverse of the exponential, i.e. $\log(\exp(V)) = V$ (see e.g. [30], [5]).

Using the Cayley map for improved efficiency and simplicity in implementation.: The exponential and its inverse are a standard (and natural) choice for differential calculus on $SE(3)$. An alternative choice is the Cayley map $\mathrm{cay}: \mathbb{R}^6 \to SE(3)$ defined (see e.g. [26]) by

$$\mathrm{cay}(V) = \begin{bmatrix} I_3 + \frac{4}{4+\|\omega\|^2}\left(\widehat{\omega} + \frac{\widehat{\omega}^2}{2}\right) & \frac{2}{4+\|\omega\|^2}(2I_3 + \widehat{\omega})\,\nu \\ 0 & 1 \end{bmatrix}, \tag{18}$$

since it is an accurate and efficient approximation of the exponential map, i.e. $\mathrm{cay}(V) = \exp(V) + O(\|V\|^3)$, it preserves the group structure, and has particularly simple to compute derivatives. Its inverse is denoted by $\mathrm{cay}^{-1} : SE(3) \to \mathbb{R}^6$ and is defined for a given $g = (R, p)$, with $R \neq -I$, by

$$\mathrm{cay}^{-1}(g) = \begin{bmatrix} [-2(I+R)^{-1}(I-R)]^\vee \\ (I+R)^{-1} p \end{bmatrix}.$$

[1] The space $T_g SE(3)$ is the tangent space on at a point $g \in SE(3)$ while $T_g^* SE(3)$ is the cotangent space of linear functions.

V. Application to Dynamic Systems

Consider a rigid body with state $x = (g, V) \in SE(3) \times \mathbb{R}^6$ with configuration

$$g = \begin{bmatrix} R & p \\ 0 & 1 \end{bmatrix}, \qquad g^{-1} = \begin{bmatrix} R^T & -R^T p \\ 0 & 1 \end{bmatrix}$$

and velocity $V = (\omega, \nu)$. Assume that the system is actuated by control inputs $u \in \mathbb{R}^c$, where $c > 1$. The dynamics is

$$\dot{g}(t) = g(t)\widehat{V(t)}, \tag{19}$$

$$\dot{V}(t) = F(g(t), V(t), u(t), w(t)), \tag{20}$$

where F encodes any Coriolis/centripetal forces, and forces such as damping, friction, or gravity, as well as the effect of control forces u. For instance, simple models of aerial or underwater vehicles assume the form

$$F(g,V,u,w) = M^{-1}\left(\mathrm{ad}_V^T M V - H(V)V + \begin{bmatrix} 0 \\ R^T f_g \end{bmatrix} + Gu + w\right),$$

where $M > 0$ is the mass matrix, $H(V) \geqslant 0$ is a drag matrix, $f_g \in \mathbb{R}^3$ is a spatial gravity force, G is a constant control transformation matrix. In this example, the uncertainty w appears as a forcing term. We employ a Lie group integrator to construct a discrete-time version which takes the form

$$g_{i+1} = g_i \mathrm{cay}(\Delta t_i V_{i+1}), \tag{21}$$

$$V_{i+1} = F_i(g_i, V_i, u_i, w_i), \tag{22}$$

where cay is the Cayley map defined in (18), $\Delta t_i \triangleq t_{i+1} - t_i$, and F_i encodes a numerical scheme for computing V_{i+1}. For instance, a simple first-order Euler step corresponds to setting

$$F_i(g_i, V_i, u_i, w_i) = V_i + \Delta t_i F(g_i, V_i, u_i, w_i),$$

while in a second-order accurate trapezoidal discretization, F_i corresponds to the implicit solution of

$$V_{i+1} = V_i + \frac{1}{2}\left[\Delta t_i F(g_i, V_i, u_i, w_i) + \Delta t_{i+1} F(g_i, V_{i+1}, u_i, w_i)\right],$$

which in general is solved using an iterative procedure such as Newton's method.

Estimation Cost.: Consider the optimal estimation of the vehicle trajectory using measurements of its velocity as

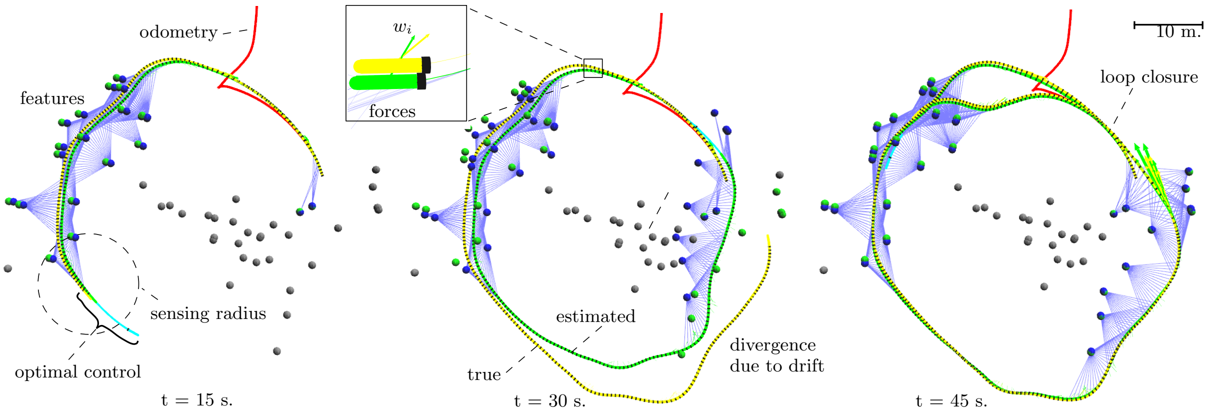


Fig. 2. Receding horizon optimal control and optimal estimation of vehicle trajectory and environmental landmarks implemented using the same algorithm: parameter-dependent DDP. MPC is formulated as optimization over the future controls u_i while estimation over the unknown past disturbance forces w_i.

well as 3-d landmark positions using e.g. a stereo camera. Denote the indices of all observed landmarks at time t_i by the set O_i. A measurement at time t_i then contains

$$z_i = (\widetilde{V}_i, \{\tilde{r}_{ij}\}_{j\in O_i}),$$

where $\widetilde{V}_i$ is the measured velocity (e.g. from an IMU and odometry) and $\tilde{r}_{ij}$ denotes the observation of landmark j from pose i. The stage-wise costs at stage i are

$$\begin{aligned} L_i(x_i, w_i, \rho) = &\frac{1}{2}\|w_i\|^2_{\Sigma^{-1}_{w_i}} + \frac{1}{2}\|V_i - \widetilde{V}_i\|^2_{\Sigma^{-1}_{V_i}} \\ &+ \sum_{j\in O_i} \underbrace{\frac{1}{2}\|R_i^T(\ell_j - p_i) - \tilde{r}_{ij}\|^2_{\Sigma^{-1}_{r_{ij}}}}_{\triangleq e(g_i, \ell_j|\tilde{r}_{ij})}. \end{aligned} \tag{23}$$

The gradient and Hessian of L_i with respect to V and ω are straightforward to compute, and with respect to g are defined with the help of the trivialized gradient (15). In particular, the derivatives of the last term in (23) are given by

$$d_g e = \begin{pmatrix} -\hat{r}y \\ -y \end{pmatrix}, \quad d_g^2 e = \begin{bmatrix} \hat{r}^T\Sigma_r^{-1}\hat{r} + \frac{\hat{r}\hat{y}}{2} + \frac{\hat{y}\hat{r}}{2} & \Sigma_r^{-1}\hat{r} - \frac{\hat{y}}{2} \\ (\Sigma_r^{-1}\hat{r} - \frac{\hat{y}}{2})^T & \Sigma_v^{-1} \end{bmatrix},$$

$$\nabla_\ell e = Ry, \qquad \nabla^2_\ell e = R\Sigma_r^{-1}R^T,$$

$$\nabla_\ell d_g e = \begin{bmatrix} R(\Sigma_r^{-1}\hat{r} - \widehat{y}) & -R\Sigma_r^{-1} \end{bmatrix},$$

where $r \triangleq R^T(\ell - p)$, $y \triangleq \Sigma_r^{-1}(r - \tilde{r})$, and all quantities are defined for the r_{ij}-th measurement.

Control Cost.: For control purposes we consider the stage-wise cost

$$L_i(x_i, u_i, \rho) = \frac{1}{2}\|V_i - V_d\|^2_{Q_{V_i}} + \frac{1}{2}\|u_i\|^2_{R_i},$$

and terminal cost

$$L_N(x_N, \rho) = \frac{1}{2}\|\text{cay}^{-1}(g_f^{-1}g_N)\|^2_{Q_{g_f}} + \frac{1}{2}\|V_N - V_f\|^2_{Q_{V_f}},$$

and where $Q_{V_i} \geqslant 0, Q_{g_i} \geqslant 0, R_i > 0$ are appropriately chosen diagonal matrices to tune the vehicle behavior while reaching a desired final state $x_f = (g_f, V_f)$. The derivatives of L_i are straightforward to compute. Only the derivatives of L_N depend on g and involve the Cayley map. They are given by

$$\begin{aligned} d_g L_N(x_N, \rho) &= (\text{dcay}^{-1}(-\Delta_N))^T Q_{g_f}\Delta_N, \\ d_g^2 L_N(x_N, \rho) &\approx (\text{dcay}^{-1}(-\Delta_N))^T Q_{g_f}\text{dcay}^{-1}(-\Delta_N), \end{aligned}$$

where $\Delta_N = \text{cay}^{-1}(g_f^{-1}g_N)$. Note that the Hessian can be approximated by ignoring the second derivative of cay as long as the vehicle can truly reach near g_f.

The trivialized Cayley derivative denoted by $\text{dcay}(V)$ for some $V = (\omega, \nu) \in \mathbb{R}^6$ is defined (see e.g. [26]) as

$$\text{dcay}(V) = \begin{bmatrix} \frac{2}{4+\|\omega\|^2}(2I_3 + \widehat{\omega}) & 0_3 \\ \frac{1}{4+\|\omega\|^2}\widehat{\nu}(2I_3 + \widehat{\omega}) & I_3 + \frac{1}{4+\|\omega\|^2}(2\widehat{\omega} + \widehat{\omega}^2) \end{bmatrix}, \tag{24}$$

it is invertible and its inverse has the simple form

$$\text{dcay}^{-1}(V) = \begin{bmatrix} I_3 - \frac{1}{2}\widehat{\omega} + \frac{1}{4}\omega\omega^T & 0_3 \\ -\frac{1}{2}\left(I_3 - \frac{1}{2}\widehat{\omega}\right)\widehat{\nu} & I_3 - \frac{1}{2}\widehat{\omega} \end{bmatrix}. \tag{25}$$

Linearization of dynamics.: The final step is to provide the linearization of the dynamics in the form (17), where $dx_i = (dg_i, \delta V_i)$. The linearization matrices are given (we omit the details for brevity) by

$$A_i = \begin{bmatrix} \text{Ad}_{\text{cay}(-\Delta t_i V_{i+1})} & \Delta t_i \text{dcay}(-\Delta t_i V_{i+1}) \\ 0 & I \end{bmatrix} \begin{bmatrix} I & 0 \\ d_g F & \nabla_V F \end{bmatrix}$$

$$B_i = \begin{bmatrix} \text{Ad}_{\text{cay}(-\Delta t_i V_{i+1})} & \Delta t_i \text{dcay}(-\Delta t_i V_{i+1}) \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 \\ \nabla_\xi F \end{bmatrix},$$

where $\xi \triangleq u$ for control, and $\xi \triangleq w$ for estimation problems.

In a preliminary study, the algorithm is applied to the parameter estimation and environmental mapping of a simulated autonomous underwater vehicle (AUV) with a simple second-order planar underactuated rigid body model with two differential thrusters and unknown linear drag terms. The parameters are $\rho = (d, \ell_1, \ell_2, \ldots, \ell_M)$, where $d \in \mathbb{R}^3$ are the three damping terms for each degree of freedom, and $\ell_j \in \mathbb{R}^2$ denote landmark locations. The vehicle is also subject to external forces modeled as disturbances $w(t)$. The vehicle executes a "figure-8" path and observes 300 landmarks but due to uncertainties in its model and measurements has a poor estimate of the path traveled (Figure 1). The DDP algorithm is executed over the whole past horizon to correct the estimate after 8 iterations. In this setting the optimization is over the parameters ρ and uncertain forces w_i since the controls u_i are known. Figure 2 shows the same approach applied with combination with MPC implemented using the same DDP algorithm.

VI. Conclusion

This paper considers the solution of estimation and control problems through a common optimization-based approach. Our key motivation is the unified and systematic treatment of dynamics, control, and sensing constraints during both estimation and control in order to increase the performance of autonomous systems such as agile robotic vehicles. As a particular computational solution, we developed parameter-dependent version of differential dynamic programming, which is a well-established method for optimal control. This paper demonstrated that DDP can be employed for both estimation and control problems using different cost functions, and by optimizing over unknown or uncertain forces during estimation and optimizing over control inputs during control. The method was implemented in simulation using rigid-body models applicable to underwater or aerial vehicles. Several key issues remain to be studied in order establish the exact computational advantages of the proposed method. In particular, we expect that the method should be most advantageous when the number of static parameters is small and must be used in a receding horizon fashion. In the near future, we will also implement the method on real systems and analyze the benefits of exploiting dynamics and optimization over forces in comparison to standard SLAM techniques based on kinematic or unconstrained models.

Acknowledgments

We thank Matthew Sheckells for correcting the formula for the Hessian of the estimation cost in §V.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *Intl. Conf. on Computer Vision (ICCV)*, 2009.

[2] D. P. Bertsekas. *Nonlinear Programming, 2nd ed.* Athena Scientific, Belmont, MA, 2003.

[3] John Betts. Survey of numerical methods for trajectory optimization. *Journal of Guidance, Control, and Dynamics*, 21(2):193–207, 1998.

[4] A. E. Bryson and Y. C. Ho. *Applied Optimal Control*. Blaisdell, New York, 1969.

[5] Francesco Bullo and Andrew Lewis. *Geometric Control of Mechanical Systems*. Springer, 2004.

[6] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. SfM with MRFs: Discrete-continuous optimization for large-scale structure from motion. *IEEE Trans. Pattern Anal. Machine Intell.*, 2012.

[7] John L Crassidis and John L Junkins. *Optimal Estimation of Dynamic Systems*. Chapman and Hall, Abingdon, 2004.

[8] M. Cummins and P. Newman. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Intl. J. of Robotics Research*, 27(6):647–665, June 2008.

[9] M. Cummins and P. Newman. Highly scalable appearance-only slam - fab-map 2.0. 2009.

[10] A.J. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Machine Intell.*, 29(6):1052–1067, Jun 2007.

[11] F. Dellaert, J. Carlson, V. Ila, K. Ni, and C.E. Thorpe. Subgraph-preconditioned conjugate gradient for large scale slam. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[12] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, Dec 2006.

[13] Frank Dellaert. Factor graphs and GTSAM: A hands-on introduction. Technical Report GT-RIM-CP&R-2012-002, Georgia Institute of Technology, 2012.

[14] H.F. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics & Automation Magazine*, Jun 2006.

[15] Garry A. Einicke. *Smoothing, Filtering and Prediction: Estimating the Past, Present and Future*. InTech, 2012.

[16] P. J. Enright and B. A. Conway. Discrete approximations to optimal trajectories using direct transcription and nonlinear programming. In *Astrodynamics 1989*, pages 771–785, 1990.

[17] R.M. Eustice, H. Singh, and J.J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Trans. Robotics*, 22(6):1100–1114, Dec 2006.

[18] Rolf Findeisen Frank Allgower and Zoltan K. Nagy. Nonlinear model predictive control: From theory to application. *J. Chin. Inst. Chem. Engrs*, 35(3):299–315, 2004.

[19] U. Frese. A proof for the approximate sparsity of SLAM information matrices. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 331–337, 2005.

[20] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control: Theory and Algorithms*. Communications and Control Engineering. Springer, 1st ed. edition, 2011.

[21] David H. Jacobson and David Q. Mayne. *Differential dynamic programming*. Modern analytic and computational methods in science and mathematics. Elsevier, New York, 1970.

[22] Y.-D. Jian, D. Balcan, and F. Dellaert. Generalized subgraph preconditioners for large-scale bundle adjustment. In *Intl. Conf. on Computer Vision (ICCV)*, 2011.

[23] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31:217–236, Feb 2012.

[24] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.

[25] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, 2009.

[26] M. Kobilarov and J. Marsden. Discrete geometric optimal control on Lie groups. *IEEE Transactions on Robotics*, 27(4):641–655, 2011.

[27] L.-Z. Liao and C.A Shoemaker. Convergence in unconstrained discrete-time differential dynamic programming. *Automatic Control, IEEE Transactions on*, 36(6):692–706, Jun 1991.

[28] D.Q. Mayne and H. Michalska. Receding horizon control of nonlinear systems. *Automatic Control, IEEE Transactions on*, 35(7):814–824, 1990.

[29] E. Mizutani and Hubert L. Dreyfus. Stagewise newton, differential dynamic programming, and neighboring optimum control for neural-network learning. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1331–1336 vol. 2, 2005.

[30] Richard M. Murray, Zexiang Li, and S. Shankar Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC, 1994.

[31] R.A. Newcombe, S.J. Lovegrove, and A.J. Davison. DTAM: Dense tracking and mapping in real-time. In *Intl. Conf. on Computer Vision (ICCV)*, pages 2320–2327, Barcelona, Spain, Nov 2011.

[32] Paul M. Newman and John J. Leonard. Consistent convergent constant time SLAM. In *Intl. Joint Conf. on AI (IJCAI)*, 2003.

[33] K. Ni, D. Steedly, and F. Dellaert. Tectonic SAM: Exact; out-of-core; submap-based SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Rome; Italy, April 2007.

[34] Kai Ni and Frank Dellaert. Multi-level submap based slam using nested dissection. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[35] G. Sibley, C. Mei, I. Reid, and P. Newman. Adaptive relative bundle adjustment. In *Robotics: Science and Systems (RSS)*, 2009.

[36] G. Sibley, C. Mei, I. Reid, and P. Newman. Vast scale outdoor navigation using adaptive relative bundle adjustment. *Intl. J. of Robotics Research*, 29(8):958–980, 2010.

[37] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Intl. J. of Robotics Research*, 23(7-8):693–716, 2004.

[38] E. Todorov. Optimality principles in sensorimotor control. *Nature neuroscience*, 7(9):907–915, 2004.

[39] E. Todorov. General duality between optimal control and estimation. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 4286–4292. IEEE, 2008.