Stagewise Newton Method for Dynamic Game Control with Imperfect State Observation

Armand Jordana, Bilal Hammoud, Justin Carpentier, Ludovic Righetti

To cite this version:

Armand Jordana, Bilal Hammoud, Justin Carpentier, Ludovic Righetti. Stagewise Newton Method for Dynamic Game Control with Imperfect State Observation. IEEE Control Systems Letters, 2022, 10.1109/LCSYS.2022.3184657 . hal-03705557

HAL Id: hal-03705557
https://inria.hal.science/hal-03705557v1
Submitted on 27 Jun 2022
Stagewise Newton Method for Dynamic Game Control with Imperfect State Observation

Armand Jordana Student Member, IEEE, Bilal Hammoud Student Member, IEEE, Justin Carpentier Member, IEEE and Ludovic Righetti, Senior Member, IEEE

Abstract—In this letter, we study dynamic game optimal control with imperfect state observations and introduce an iterative method to find a local Nash equilibrium. The algorithm consists of an iterative procedure combining a backward recursion similar to minimax differential dynamic programming and a forward recursion resembling a risk-sensitive Kalman smoother. A coupling equation renders the resulting control dependent on the estimation. In the end, the algorithm is equivalent to a Newton step but has linear complexity in the time horizon length. Furthermore, a merit function and a line search procedure are introduced to guarantee convergence of the iterative scheme. The resulting controller reasons about uncertainty by planning for the worst case disturbances. Lastly, the low computational cost of the proposed algorithm makes it a promising approach to do output-feedback model predictive control on complex systems at high frequency. Numerical simulations on realistic robotic problems illustrate the risk-sensitive behavior of the resulting controller.

Index Terms—Game theory, Optimal control.

I. INTRODUCTION

DOING output-feedback Model Predictive Control (MPC) while being robust to the estimator uncertainty is a notoriously difficult problem [1]. In optimal control, when the state measurement is partial and corrupted by noise, a common practice is to treat the estimation problem independently from the control and rely on the most likely outcome. However, in some scenarios, one might want to design a controller robust to the worst case estimation error. Such controllers can be obtained through dynamic game control [2], [3]. As Mayne [1] advocates, one meaningful way to properly take into account the measurement uncertainty is to use a minimax formulation linking control and estimation.

This problem has been extensively studied in [4], [5], which shows that a specific dynamic game formulation leads to MPC approaches with bounded state trajectories and provides an explicit characterization of these bounds. However, the minimax problem was solved with an interior point method without taking into account the specific structure of the problem and the sparsity induced by time. In this work, we derive an explicit iterative solution that fully exploits sparsity, resulting in an algorithm that linearly scales with the time horizon length and which can be easily warm-started for use in MPC schemes [6].

In the linear dynamics and quadratic cost case, Jacobson [7] showed that dynamic game control is equivalent to risk sensitive control and derived a closed form solution. Later, Whittle [8] extended the results to the linear quadratic case with imperfect state observations. In [9], a first iterative version of Whittle’s solution was introduced to tackle the nonlinear risk sensitive problem with imperfect observations. However, the stochastic nature of the problem hindered the development of theoretical guarantees. Although, dynamic game and risk sensitive control are equivalent in the linear quadratic case [7], this is no longer the case in the nonlinear setting [10]. Nonetheless, dynamic game control is tightly connected to robust and risk sensitive control. In [2], [10], James and Campi showed that dynamic game control can be interpreted as the limit case of risk sensitive control when noise tends to zero. Recently, Başar [11] presented a detailed overview of the connections between both problems in continuous time. Additionally, Başar and Bernhard [3], [12] established the connections between dynamic game control and $H_{\infty}$-Optimal Control both in the perfect and imperfect state information case.

For nonlinear systems, estimating a state trajectory corresponding to some given measurements is usually intractable analytically. A common approach is to model the noise as Gaussian and to maximize the Maximum A Posteriori (MAP) [13]. If the dynamics are affine, then the problem can be solved analytically with the so-called Rauch–Tung–Striebel (RTS) smoother [14]. The RTS smoother is made of a forward recursion which resembles the Kalman Filter (KF) and a backward recursion. In the nonlinear case, iterative schemes are usually used. A popular choice is the iterative Kalman smoother which is equivalent to a Gauss-Newton method on the MAP [15]. The estimation part of our proposed solution resembles a risk sensitive version of this smoother.

In optimal control or dynamic game control with perfect state information, various numerical optimization algorithms
have been developed to iteratively find solutions in the non-
linear case. In optimal control, the most analogous to our
work are Differential Dynamic Programming (DDP) [16] and
the stagewise implementation of the Newton’s method [17].
The stagewise Newton method is an exact implementation
of Newton’s method that exploits the specific structure of
the Hessian matrix in order to scale linearly with the time
horizon. DDP is an iterative algorithm that takes an update
step on the control input by applying dynamic programming
on a quadratic approximation of the value function. In [16],
Murray showed that DDP is very similar to a Newton step and
inherits its convergence properties. For dynamic game control
with perfect state information, the seminal work from [18],
[19] introduced minimax DDP showing that DDP could be ex-
tended to zero-sum two players games. Recently, [20] further
extended the concepts of stagewise Newton method and DDP
to nonzero-sum games with an arbitrary number of players in
the full information case.

Despite having been widely studied theoretically, to the
best of our knowledge, dynamic game control with imperfect
state observations has not been approached from a numerical
optimization point of view. In this work, we consider the
general problem of dynamic game control with imperfect
state observation and present a numerically efficient provably
convergent algorithm to solve it. The solution presented gener-
alizes commonly used techniques such as the extended Kalman
smoother and Differential Dynamic Programming. The pro-
posed solution fully exploits the sparsity of the problem and
scales linearly with time, making it a promising method for
model predictive control.

Contributions. In this paper, we derive a stagewise New-
ton method for dynamic game control with imperfect state
observations. The proposed approach scales linearly in the
time horizon. Furthermore, a merit function and a line search
procedure are introduced to guarantee convergence. To our
knowledge, this is the first iterative procedure with con-
vergence guarantee for the dynamic game control problem
with imperfect state observation. The proposed solver couples
estimation and control by merging an iterative optimal control
algorithm similar to minimax DDP and an iterative risk
sensitive Kalman smoother. We illustrate the behavior of the
resulting controller on two realistic robotic problems.

Notations. The derivative of a function $f$ with respect
to a vector $v$ is denoted by $f^v$, similarly for second order
dervatives with respect to vectors $u, v$ is denoted as $f^{uv}$.
If $(v_i)_{i \in \mathbb{N}}$ is a sequence of vectors, then $v_{k:t}$ is a vector
concatenating $v_k, \ldots, v_t$. $1_{x \in A}$ is the indicator function which
equals 1 if $x \in A$ and 0 otherwise. $I_n$ denotes the identity
matrix of size $n$ by $n$.

II. PROBLEM STATEMENT

Similar to [4], [5], this work studies a special class of
nonlinear dynamic games with imperfect state observation [2].
Given a history of measurements $y_{t:t}$, a history of control
inputs, $u_{0:t-1}$ and a prior on the initial state $\hat{x}_0$, we aim
to find a control sequence $u_{t:T-1}$ that minimizes a given
cost $\ell$ while an opposing player aims to find the disturbances
$(w_{0:T}, \gamma_{1:t})$ that maximize this cost $\ell$ minus a weighed norm
of the disturbances. Such a problem is formally written as:

$$\min_{u_{t:T-1}} \max_{w_{0:T}} \max_{\gamma_{1:t}} \sum_{j=0}^{T-1} \ell_j(x_j, u_j) + \ell_T(x_T)$$

subject to

$$x_0 = \hat{x}_0 + w_0,$$

$$x_{j+1} = f_j(x_j, u_j) + w_{j+1}, \quad 0 \leq j < T,$$

$$y_j = h_j(x_j) + \gamma_j, \quad 1 \leq j \leq t.$$

where $\mu > 0$. $x_j$ is the state, $\omega_j$ the process disturbance, $\gamma_j$ the measurement disturbance, $T$ the time horizon, $t$ the current
time. The transition model $f_j$, the measurement model $h_j$ and the cost $\ell_j$ are assumed to be $C^2$. The measurement uncertainty
$R_j$, the process uncertainty $Q_j$ and the initial state uncertainty
$P$ are positive-definite matrices.

Interestingly, this problem encompass various formulations
of control and estimation. If $t = 0$ and if $w_0$ is fixed to zero, we
recover dynamic game control with perfect state information.
Additionally, if $t = 0$, in the limit where $\mu$ tends to zero, we
find the generic optimal control formulation [10]. And lastly,
if $t = T$ and if we consider all the cost $\ell_j$ to be null, then,
(1) is equivalent to maximizing the MAP.

III. METHOD

A. First order conditions for a Nash equilibrium

The main challenge in the problem formulation (1) is the
equality constraint maintaining the dynamic feasibility. A
popular approach [19] is to derive a DDP-like algorithm by
sequentially taking quadratic approximations of the value
function recursion. However, a stagewise Newton step can readily
be derived. One of the key features of the dynamic game (1) is
that by changing the decision variable of the opposing player,
one can transform the problem into an unconstrained one [2].
Indeed, we can use the equality constraints of Eq. (2) to replace
disturbance maximization into a maximization over the state
sequence. Then the problem loses its equality constraints and
can be formulated as the search of the saddle point of

$$J(x_{0:T}, u_{t:T-1}) = \sum_{j=0}^{T-1} \ell_j(x_j, u_j) + \ell_T(x_T)$$

subject to

$$\min_{x_0} \max_{u_{t:T-1}} \max_{w_{0:T}} \sum_{j=0}^{T-1} \ell_j(x_j, u_j) + \ell_T(x_T)$$

subject to

$$x_0 = \hat{x}_0 + w_0,$$

$$x_{j+1} = f_j(x_j, u_j) + w_{j+1}, \quad 0 \leq j < T,$$

$$y_j = h_j(x_j) + \gamma_j, \quad 1 \leq j \leq t.$$
such that for any \((x_{0:T}, u_{t:T-1})\) satisfying \(\|x_{0:T} - x^*_0\| < \delta\) and \(\|u_{t:T-1} - u^*_{t:T-1}\| < \delta\), we have
\[
J(x_{0:T}, u^*_{t:T-1}) \leq J(x_{0:T}, u_{t:T-1}) \leq J(x^*_0, u_{t:T-1}). \tag{4}
\]
A standard approach to this problem is to search for a stationary point [21]:
\[
\left(\frac{\partial J}{\partial x_{0:T}}(x^*_0, u^*_{t:T-1}), \frac{\partial J}{\partial u_{t:T-1}}(x^*_0, u^*_{t:T-1})\right) = 0 \tag{5}
\]
Interestingly, the change of decision variable in Eq. 3 turned the problem into an unconstrained one but made no assumption on the structure of the cost. The only required assumption is that each disturbance is in a one to one map with the state at each time step.

B. About the Linear Quadratic case

In the linear quadratic case, Whittle has shown that under the assumption that each disturbance is in a one to one map with the state at each time step, the Newton step, denoted by \(p\), satisfies
\[
J(x_{0:T}, u^*_{t:T-1}) \leq J(x_{0:T}, u_{t:T-1}) \leq J(x^*_0, u_{t:T-1}). \tag{4}
\]
A standard approach to this problem is to search for a stationary point [21]:
\[
\left(\frac{\partial J}{\partial x_{0:T}}(x^*_0, u^*_{t:T-1}), \frac{\partial J}{\partial u_{t:T-1}}(x^*_0, u^*_{t:T-1})\right) = 0 \tag{5}
\]
Here, \(p = (p^T_{0:T}, p^T_{0:T-1})^T\) where \(p_{0:T} \in \mathbb{R}^{(T+1)n_x}\) is a stack of vectors \(p_k \in \mathbb{R}^{n_x}\) with \(n_x\) being the dimension of the state space. Similarly, \(p_{u_0:T-1} \in \mathbb{R}^{(T-1)n_u}\) is a stack of vectors \(p_k \in \mathbb{R}^{n_u}\) with \(n_u\) being the dimension of the control space.

To simplify the notation, we define an augmented Hessian of the cost that contains the second order derivatives of the dynamics for all \(k < T\):
\[
\bar{J}^x_k = \tilde{J}^x_k + \mu - 1 \leq x^*_k + 1 \geq \gamma_k T R_k h^x_k, \tag{6}
\]
where the derivatives are evaluated at the current guess and \(\gamma_k^x := y_k - h_k(x^*_k)\). Here, the second order derivatives of the dynamics are tensors. The exact definition of the tensor indexing and the tensor product is provided in the supplementary material [23].

The next three propositions are analogous to the principles introduced by Whittle: the past stress recursion, the future stress recursion and the coupling of the past and future stress recursions. The first proposition, analogous to the future stress recursion, expresses every future state and control update steps as a function of \(p_{x_k}\).

Proposition 1 (Future stress). In Equation (6), the last \((T-t)(n_x + n_u)\) rows are equivalent to:
\[
\forall k \geq t, \quad p_{uk} = G_k p_{x_k} + g_k \tag{8}
\]
where \(V_k\) and \(v_k\) are solutions of the backward recursion:
\[
\Gamma_{k+1} = I - \mu V_{k+1} Q_{k+1}\tag{9}
\]
with the terminal condition
\[
V_T = \tilde{J}^x_T, \quad v_T = \tilde{J}^x_T. \tag{10}
\]
In those equations, \(\mu\) intervenes only in \(\Gamma_k\) and the augmented terms of the cost. Interestingly, \(\Gamma_{k+1}\) shifts at each time step, the value function terms \(V_k\) and \(v_k\). Then, the second proposition, analogous to the past stress recursion, expresses every past state update steps as a function of \(p_{x_k}\).

Proposition 2 (Past stress). In Equation (6), if \(t \geq 1\), the first \((t - 1)n_x\) rows are equivalent to: \(\forall k = 0, \ldots, t-1,\)
\[
p_{x_k} = E_k^{-1} \left( f_k^T Q_{k+1}(w_{k+1} + p_{x_{k+1}}) + P_k^{-1} \mu_k + \mu^T_k \right) \tag{11}
\]
where $P_k$ and $\mu_k$ are solution of the forward recursion:

$$
E_{k+1} = P_k^{-1} + f_k^T Q_k^{-1} f_k - \mu_k^2 f_k \\
\tilde{P}_{k+1} = Q_{k+1} + f_k^T (P_k - \mu_k^2 f_k^{-1}) f_k^T \\
K_{k+1} = \tilde{P}_{k+1} h_{k+1} (R_{k+1} + h_{k+1}^T \tilde{P}_{k+1} h_{k+1}^{-1})^{-1} \\
P_{k+1} = (I - K_{k+1} h_{k+1}^T) \tilde{P}_{k+1} \\
\mu_{k+1} = (I - K_{k+1} h_{k+1}^T) (f_k^T \mu_k - w_{k+1}^i) + K_{k+1} \gamma_{k+1} \\
+ \mu P_{k+1} Q_{k+1} f_k^T E_{k+1}^{-1} (f_k^T \tilde{\mu} + \ell_k)$$  \hspace{1cm} (12)

with the initialization

$$P_0 = P, \quad \mu_0 = \tilde{x}_0 - x_0^i.$$  \hspace{1cm} (13)

Interestingly, if all the cost terms $\ell_j$ are zero and if $t = T$, the method is equivalent to a Newton method on the MAP. By construction, the expected decrease of the derivations of each gradient are provided in the supplementary material [23].

### Proposition 3 (Coupling).

In Equation (6), the remaining rows (from $tn_x + 1$ to $(t+1)n_x$) are equivalent to

$$P_{x^i} = (P_k^{-1} - \mu V_i)^{-1} (P_k^{-1} \mu + \mu V_i).$$  \hspace{1cm} (14)

In the limit case when $\mu$ tends to zero, the estimation and control are decoupled and we recover the usual certainty equivalence principle: $p_{x^i} = \hat{x}_t$. The algorithm is then equivalent to an iterative estimator and an iterative controller running independently. More precisely, at each iteration the controller uses the current estimate of the smoother.

**Proof.** The proof follows from the analytical derivations of the gradient and the Hessian of (3), a forward recursion from to an iterative estimator and an iterative controller running independently. More precisely, at each iteration the controller uses the current estimate of the smoother. □

**Algorithm 1:** Stagewise Newton step

- **Input:** $x_0^i, x_1^i, \ldots, x_T^i, u_0^i, \ldots, u_{T-1}^i$
- **Estimation forward pass**
  1. $P_k \leftarrow P_k, \mu_k \leftarrow \tilde{x}_0 - x_0^i$  
  2. for $k = 0, \ldots, t - 1$
    3.    $P_{k+1}, \mu_{k+1} \leftarrow \text{Eq. (12)}$
  4. $V_T \leftarrow \ell_T^e, \mu_T \leftarrow \ell_T^e$
  5. for $k = T - 1, \ldots, t$
    6.    $V_k, u_k \leftarrow \text{Eq. (9)}$
  7. $p_{x^i} \leftarrow (P_k^{-1} - \mu V_k)^{-1} (P_k^{-1} \mu + \mu V_k)$
  8. **Estimation backward pass**
  9. for $k = t - 1, \ldots, 0$
    10.    $p_{u_k}, p_{x_k} \leftarrow \text{Eq. (11)}$
  11. **Control forward pass**
  12. for $k = t, \ldots, T - 1$
    12.    $p_{u_k}, p_{x_k} \leftarrow \text{Eq. (8)}$

**Output:** $p_{x_0}, p_{x_1}, \ldots, p_{x_T}, p_{u_0}, \ldots, p_{u_{T-1}}$

In the end, the update step, $p$, can be computed with a forward recursion on the past indexes, a backward recursion on the future indexes, a coupling equation, a backward recursion on the past indexes and a forward recursion on the future indexes. Algorithm 1 summarizes those steps. Clearly, the complexity is linear in time. Instead of inverting a matrix of size $((T + 1)n_x + (T - t)n_u)$, Algorithm 1 only operates with matrices of size $n_x$ or $n_u$ and the number of operation is proportional to $T$.

### D. Line-search and convergence

In the linear quadratic case, Algorithm 1 is equivalent to Whittle’s derivations and only one iteration is required to find a solution. However, in the general nonlinear case, several iterations of Newton’s step are required. A common approach to guarantee the convergence of the overall iterative procedure is to introduce a line search and a merit function [22]. Given a guess at iteration $i$ and a direction $p$, the next guess is defined by

$$x_{i+1}^{0:T} = x_i^{0:T} + \alpha_i \left( \frac{P_{x_{i:T}}}{p_{x_{i:T}}}, u_{i:T-1} \right),$$  \hspace{1cm} (15)

where the step length $\alpha_i$ is chosen in order to decrease the merit function. As advocated by Nocedal et al. [22], a Newton step provides a descent direction for the merit function

$$f_M(x_{0:T}, u_{0:T-1}) = \frac{1}{2} \sum_{j=0}^{T} \left\| \frac{\partial J}{\partial x_j} \right\|^2 + \frac{1}{2} \sum_{j=0}^{T-1} \left\| \frac{\partial J}{\partial u_j} \right\|^2.$$  \hspace{1cm} (16)

This merit function can be derived analytically. The exact derivations of each gradient are provided in the supplementary material [23]. By construction, the expected decrease of the direction $p$ derived by Algorithm 1 is $p^T \nabla f_M = -\|\nabla J\|^2$ and the following convergence guarantees hold.

**Proposition 4.** Assuming that the norm of the inverse of the Hessian of $J$ is bounded and that the step length $\alpha_i$ satisfies the Wolfe conditions for the merit function (16), the sequence $(x_{i:T}^{0:T}, u_{i:T-1}^{0:T})$, defined by the update rule (15) with steps from Algorithm 1 is globally convergent to a stationary point of (3). Furthermore, when the iterate is sufficiently close to the solution, the sequence has quadratic convergence.

**Proof.** This proposition is a direct consequence from the fact that Algorithm 1 yields a Newton step. A detailed proof of the convergence guarantees of Newton method is provided in [22].

One may ask under which conditions the Hessian of $J$ is non-degenerate. Intuitively, large values of $\mu$ can make the problem ill-defined. Indeed, if the opposing player can choose large disturbances, then the controller might not be able to compensate. In the linear quadratic case, Whittle [8] studied this maximum value for $\mu$ that makes the problem ill-defined. Although, we do not study this limit value in the nonlinear case, we note that analogously to the linear quadratic case, the algorithm is well defined if $\Gamma_k$ and $P_k^{-1} - \mu \ell_k^2$ are positive-definite which is the case when $\mu$ is small enough.

### E. About the cooperative case

So far, only the case $\mu > 0$ has been considered, but, the case $\mu < 0$ is also well defined. Indeed, the search for
a stationary point of \( J \) can be done for an arbitrary sign of \( \mu \). However, such a stationary point would now be a way to find a local minimum of \( J \) with respect to the variables \((x_0; T, u_t; t-1)\). Interestingly, this scenario can be interpreted as a cooperative scenario between the controller and the opposing player. In fact, the disturbances can be seen a second controller minimizing the cost, \( \ell \), and maximizing the likelihood of the disturbances. Clearly, this change of sign does not affect the derivation of the stagewise Newton method. However, it can be noted that in that case, one can directly use the cost \( J \) as a merit function.

### IV. Experiments

A Python implementation of the proposed method is available online [23]. It is based on the Crocoddyl software [6], a state of the art (risk-neutral) DDP solver which provides analytical derivatives of robot dynamics. In this section, two numerical examples illustrate the proposed method.

#### A. Planar quadrotor

We study a quadrotor moving in a plane aiming to reach the position \((p_x, p_y) = (2, 0)\) starting at the origin without initial velocity. The state is \( x = (p_x, p_y, \theta, p_x, p_y, \theta)^T \) where \( \theta \) is the orientation of the quadrotor. The system dynamics is

\[
\begin{align*}
\dot{p}_x &= -(u_1 + u_2) \sin(\theta), \\
\dot{p}_y &= (u_1 + u_2) \cos(\theta) - mg, \\
\dot{\theta} &= r(u_1 - u_2),
\end{align*}
\]

where the control input \( u = (u_1, u_2)^T \in \mathbb{R}^2 \) represents the force at each rotor and \( g \) is gravitational acceleration. An exponential cost models the presence of an obstacle

\[
\ell_k(x_k, u_k) = 0.3 \exp\left(-10(p_{x_k} - 1)^2 - 0.5(p_{y_k} + 1)^2\right) + 0.005 \|u_k - \bar{u}\|^2 + 0.05(x_k - x_*)^T L(x_k - x_*)
\]

\[
\ell_T(x_T) = (x_T - x_*)^T L(x_T - x_*)
\]

where \( \bar{u} = \frac{mg}{2}(1, 1)^T \), \( x_* = (2, 0, 0, 0, 0, 0)^T \), and \( L = \text{diag}(100, 100, 100, 1, 1, 1) \). Only the position, \( p_x, p_y \), and orientation \( \theta \) are part of the measurements. The integration and discretization of the model is done with Runge Kutta 4 with a time step of 0.05 and the total horizon, \( T \), is 60. Furthermore, \( P_0 \approx Q = 10^{-5}I_6, R = 10^{-4} \text{diag}(1, 1, 0.01) \), and \( \bar{x}_0 \) is the origin. A backtracking line-search is used – a step is accepted if \( f_M^{t+1} \leq \alpha f_M^t + \alpha c f_M^t \) where \( f_M^t = f_M(x_{0:T}, u_{t:T-1}) \) with \( c = \frac{1}{4} \). The iterative process is stopped when the decrease in the merit function is lower than \( 10^{-12} \).

Fig. 1 illustrates the solution obtained by the solver for different values of \( \mu \). The neutral controller, the limit when \( \mu \) tends to zero, aiming at minimizing the cost without accounting for disturbances, is solved using DDP. Interestingly, when \( \mu \) is positive (the non-cooperative case), the opposing player chooses disturbances that will push the quadrotor towards the obstacle and when \( \mu \) is negative (the cooperative case), the opposing player chooses disturbances that will push the quadrotor away from the obstacle. For this experiment, we found that the value of \( \mu \) for which \( \Gamma_k \) and \( P_k^{-1} - \mu E_k \) are no longer positive-definite matrices was around 20.

Fig. 2. Average trajectory. Compared the neutral controller, the dynamic game controller \( (\mu = 6) \) exhibits a risk sensitive behavior as it remains further from the high cost area representing the obstacle.

Fig. 3 shows the distribution of control trajectories. Interestingly, the dynamic game controller has a larger standard deviation.

Fig. 3. Average control trajectories. Compared the neutral controller, the dynamic game controller \( (\mu = 6) \) has a larger standard deviation.
B. An industrial robot

In this example, we consider the 7-DoF torque-controlled KUKA LWR iiwa R820 14. The dynamics of the robot is composed of the joint positions and velocities. We consider the following prior on the initial condition $\hat{x}_0 = (0.1 \ 0.7 \ 0.7 -0.5 \ 1.5 \ 0.)^T$. The control input is a 7-dimensional vector of the torque applied on each joint. The goal is to move the end effector to a desired position, $p_{\text{target}} = (-0.4 \ 0.3 \ 0.7)$ with the following cost:

$$
\ell_k(x_k, u_k) = 10^{-3} ||x_k - \hat{x}_0||^2 + 10^{-6} ||u_k - \bar{u}(x_k)||^2 + 10^{-1} ||p_{\text{target}} - \bar{p}(x_k)||^2
$$

$$
\ell_T(x_T) = ||p_{\text{target}} - \bar{p}(x_k)||^2 + 10^{-3} ||x_k - \hat{x}_0||^2, \quad (19)
$$

where $\bar{u}(x_k)$ is the gravity compensation and $\bar{p}(x_k)$ the position of the end-effector. For the measurement model, we assume that only the joints position are measured. The initial control inputs $u_0, \ldots, u_{t-1}$ are generated with DDP. Given those $t$ initial control inputs, $t$ measurements are generated according to an undisturbed trajectory. More precisely, for $1 \leq k \leq t$, the observations are defined by $y_k = h_k(f(k)(\hat{x}_0, u_1, \ldots, u_k))$ where $f(k)(\hat{x}_0, u_1, \ldots, u_k)$ denotes the state value at the $k$th step integrated from the initial guess $\hat{x}_0$. The horizon, $T$, is 100 and $t$ is 5 while the sensitivity parameter is set to $\mu = 1.2$. Here $P_0 = Q = 0.01 \times I_{14}$ and $R = 0.5 \times I_7$. For this experiment, the second order derivatives of the dynamics were approximated to be null as the former are not provided by most standard rigid body dynamics libraries such as Pinocchio [24]. Note that this is a common practice for state of the art optimal control algorithms in robotics [6]. Our solver converges nevertheless, suggesting that this approximation might be used to scale to a large number of degrees of freedom for realtime computations (e.g. MPC). In Fig. 4, we plot the solution of the dynamics game solver compared to DDP in the end effector space, the shaded grey area represents the estimation part of the solution. We can see that the dynamic game controller plans that the disturbances will slow down the closing task.

V. CONCLUSION

We introduced an iterative solver to find local Nash equilibrium of dynamic game with imperfect state measurements. The proposed algorithm is proven to be equivalent to Newton’s method and benefits from its convergence properties while scaling linearly with the time horizon.

REFERENCES

[1] D. Q. Mayne, “Model predictive control: Recent developments and future promise,” Automatica, vol. 50, pp. 2967–2986, 2014.
[2] M. R. James, J. S. Baras, and R. J. Elliott, “Risk-sensitive control and dynamic games for partially observed discrete-time nonlinear systems,” IEEE transactions on automatic control, vol. 39, pp. 780–792, 1994.
[3] T. Başar and P. Bernhard, “H∞-optimal control and related minimax design problems: A dynamic game approach,” IEEE Trans. Autom. Control, vol. 41, 1996.
[4] D. A. Copp and J. P. Hespanha, “Nonlinear output-feedback model predictive control with moving horizon estimation,” in 53rd IEEE Conference on Decision and Control. IEEE, 2014, pp. 3511–3517.
[5] T. Başar, “Simultaneous non-linear model predictive control and state estimation,” Automatica, vol. 77, pp. 143–154, 2017.
[6] C. Mantastili, R. Badhiroja, W. Merkt, G. Saurel, B. Hammod, M. Naveau, J. Capringer, L. Righetti, S. Vijayakumar, and N. Mansard, “Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control,” in IEEE International Conference on Robotics and Automation (ICRA), 2020.
[7] D. Jacobson, “Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games,” IEEE Transactions on Automatic control, vol. 18, pp. 124–131, 1973.
[8] P. Whittle, “Risk-sensitive linear/quadratic/gaussian control,” Advances in Applied Probability, vol. 13, pp. 764–777, 1981.
[9] B. Hammod, A. Jordana, and L. Righetti, “Iris: Iterative risk sensitive control for nonlinear systems with imperfect observations,” arXiv preprint arXiv:2110.06700, 2021.
[10] M. C. Campi and M. R. James, “Nonlinear discrete-time risk-sensitive optimal control,” International Journal of Robust and Nonlinear Control, vol. 6, pp. 1–19, 1996.
[11] T. Başar, “Robust designs through risk sensitivity: An overview,” Journal of Systems Science and Complexity, vol. 34, pp. 1634–1665, 2021.
[12] P. Bernhard, “Minimax versus stochastic partial information control,” in Proceedings of 1994 33rd IEEE Conference on Decision and Control, vol. 3, 1994, pp. 2572–2577 vol.3.
[13] L. Cox, “On the estimation of state variables and parameters for noisy dynamical systems,” IEEE Transactions on automatic control, vol. 9, pp. 5–12, 1964.
[14] H. E. Rauch, F. Tung, and C. T. Striebel, “Maximum likelihood estimates of linear dynamical systems,” AIAA journal, vol. 3, pp. 1445–1450, 1965.
[15] B. M. Bell, “The iterated kalman smoother as a gauss–newton method,” SIAM Journal on Optimization, vol. 4, pp. 626–636, 1994.
[16] D. Murray and S. Yakowitz, “Differential dynamic programming and newton’s method for discrete optimal control problems,” Journal of Optimization Theory and Applications, vol. 43, pp. 395–414, 1984.
[17] J. C. Dunn and D. P. Bertsekas, “Efficient dynamic programming implementations of newton’s method for unconstrained optimal control problems,” Journal of Optimization Theory and Applications, vol. 63, pp. 23–38, 1989.
[18] H. Mukai, A. Tanikawa, I. Tunay, I. Ozcak, I. Katz, H. Schattler, P. Rinaldi, G. Wang, L. Yang, and Y. Sawada, “Game-theoretic linear quadratic quadratic method for air mission control,” in 39th IEEE Conference on Decision and Control, vol. 3, 2000, pp. 2574–2580.
[19] I. Morimoto, G. Zeglin, and C. Atkeson, “Minimax differential dynamic programming: application to a biped walking robot,” in Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), vol. 2, 2003, pp. 1927–1932.
[20] B. Di and A. Lamperski, “Newton’s method, bellman recursion and differential dynamic programming for unconstrained nonlinear dynamic games,” Dynamic Games and Applications, pp. 1–49, 2021.
[21] T. Başar and G. J. Olsder, Dynamic noncooperative game theory. SIAM, 1998.
[22] J. Nocedal and S. J. Wright, Numerical optimization. Springer, 1999.
[23] A. Jordana, B. Hammod, J. Capringer, and L. Righetti, https://github.com/machines-in-motion/dynamic_game_optimizer.
[24] J. Carpentier, F. Valenza, N. Mansard, et al., “Pinocchio: fast forward and inverse dynamics for poly-articulated systems,” https://stack-of-tasks.github.io/pinocchio, 2015–2021.

![Fig. 4. End-effector position $p_{EE}$ vs time. The dashed lines represent the target.](image-url)