2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Trajectory Generation with Dynamic Programming for End-Effector Sway Damping of Forestry Machine

Iman Jebellat[1] and Inna Sharf[2].

Abstract— **When a robot end-effector is attached to the arm via passive joints, undesirable end-effector sway will occur. In a forestry crane, such as the log-loading or harvesting machine, this sway is problematic as it hinders the efficiency and also can harm the machine and environment. Here, we tackle the sway problem of the forestry forwarder by proposing a methodology for generating anti-sway trajectories in fast maneuvers. We employ the dynamic programming algorithm, combined with a suitable linearization approach, the latter identified through a comparative study. The solution has low computational cost and provides excellent performance for residual sway damping. We demonstrate the dynamic programming solution on the virtual model of the forwarder by using a high-fidelity multibody-dynamics simulator to validate its performance. The results show our optimal trajectories can suppress the residual sway effectively to be, on average, less than 10% of the sway when using fifth order polynomial trajectories, in point-to-point maneuvers starting from rest or from initial sway conditions.**

I. Introduction

A. Background and Motivation

The need for increasing autonomy of forestry machines has become apparent over the past decade. These machines, classified as mobile robots, are massive in size, and their manipulators are comprised of what is referred to as the crane (or boom) and the end-effector. Human operators may require years of training to become skilled to operate these machines and are also exposed to substantial mental and physical loads in the harsh environment of forests [1]–[3]. Manipulating the forestry cranes, in particular, is such a complicated task that studies show operators spend more than 80% of their cabin time dedicated to crane manipulation [4].

Several forestry machines are involved in timber-harvesting operations, equipped with different types of end-effectors (EE). For machines such as the forwarder, the crane has four actuated joints, and the EE which is called the grapple, is connected to the tip of the crane (boom tip) via two passive joints, as shown in Fig. 1. These passive joints ensure a nominal vertical alignment of the grapple which naturally helps with the log loading and unloading tasks. On the other hand, the passive joints cause undesirable sway or swing of the grapple during and after crane manipulation. Dealing with this sway creates an additional burden on the operator, and it can be dangerous to the environment, the machine itself and even the operator. The residual sway, which is the swinging motion that persists after the crane reconfiguration maneuver, has a deleterious impact on the productivity since the operator may be forced to wait for the sway to dampen before attempting the next task in the operation. It may also be problematic for other autonomy goals, such as acquiring image information from a grapple-fixed camera [5]. One trivial solution to the sway problem is to move the crane slowly; however, this also affects the productivity negatively. Damping the residual sway in a fast maneuver is the focal problem addressed in this paper.

B. State of the Art

To date, very few papers have reported a solution to the sway problem in the context of timber-harvesting machines. Generally, this problem can be addressed either at the trajectory generation level or at the controller design level. In [6] and [7], anti-sway controller design is considered for a simplified problem where sway is limited to one direction only by moving a subset of the actuated joints. However, in a real scenario, all four of the crane positioning joints contribute to the sway which happens in two dimensions. The authors of [8] design nonlinear model predictive controllers for the four crane joints, but demonstrate it for a system with a simpler EE configuration than the real forwarder grapple.

The payload sway problem arises in several applications other than forestry machines, for example, in construction cranes and quadrotors carrying a slung payload. Broader research on suppressing payload oscillations has employed feedback control [8]–[11], parameter optimization [12], input shaping by impulse convolution [13]–[15], and dynamic programming [16]–[18]. Among these, we choose the dynamic programming (DP) approach since, unlike previous methods, respectively, it is used in open-loop, does not rely on convex objective functions, and generates smoother trajectories [19].

Dynamic programming (DP) is an optimization algorithm based on Bellman's principle of optimality. A standard computationally inexpensive DP formulation can be applied to a linear discrete system, while for nonlinear systems, suitable linearization is required. In [19], the dynamics equations of a gantry crane were sufficiently simple so that linearization about the equilibrium point of zero sway was used to generate DP trajectories. DP was employed to suppress the sway of a payload suspended from two manipulators in [17] and a quadrotor-slung load system in [18]. Both works, employed the linearization about the current states and inputs. The main drawback of this linearization approach in the DP framework is the need for several iterations of the DP algorithm, with the corresponding numerical simulation and linearization, to provide an acceptable level of convergence. An alternative

[1]Iman Jebellat is with the Department of Mechanical Engineering, McGill University. iman.jebellat@mail.mcgill.ca

[2]Inna Sharf is a Professor at the Department of Mechanical Engineering, McGill University. She is currently a Lead Researcher at FPInnovations. inna.sharf@fpinnovations.ca

approach employed here makes use of the linearization about a nominal trajectory, which not only has a lower computational cost but also provides a superior solution.

It is also noteworthy that passively supported EEs or slung payloads are usually not guaranteed to be free of sway when the system initiates a maneuver. Non-rest-to-rest trajectories were designed in [20], [21], by re-parameterizing the input shapers. We investigate the suitability of the DP solution under such conditions, which was not examined before.

C. Contributions

In this paper, we tackle the sway problem of a forestry machine at the trajectory generation level, which was not addressed before, for point-to-point (P2P) fast maneuvers. The other specific contributions of this paper are as follows:

- We propose a methodology which employs DP to generate anti-sway trajectories for a multi-DOF serial manipulator with a passive EE, thus considering the problem in full generality.
- We demonstrate that the linearization scheme adopted to linearize the dynamics of the sway has a significant effect on the computational cost and the DP solution. We identify linearization about the nominal trajectory to provide the best overall performance for our use case.
- We validate the performance of the DP solution on the virtual model of the forwarder machine by using a high-fidelity multibody-dynamics simulator.
- We showcase the performance of the DP solution for damping the residual sway in non-zero initial sway conditions, at no additional cost.

II. Model of Forwarder Crane and Grapple

The forwarder crane is a serial robotic arm with three revolute and one prismatic joints, shown in Fig. 1. The EE is attached to the crane via two passive joints. The EE's sway, measured by the angles of the passive joints, θ_5 and θ_6, is primarily determined by the boom tip's motion, which in turn is directly dependent on the motion of the preceding joints. The dimensions and other properties of our forwarder model are based on the Tigercat 1075C forwarder.

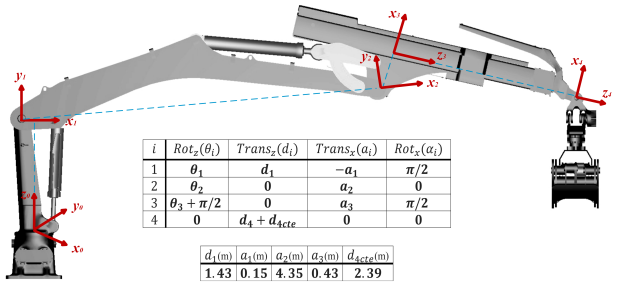
1) Crane Kinematics: Crane kinematics are defined by the DH parameters (see Fig. 1), following the convention used in [22]. The Cartesian coordinates of the boom tip ($_{bt}$) in the global frame (F_0) are:

$$\begin{aligned} x_{bt} &= \cos\theta_1(-a_1 + a_2\cos\theta_2 - a_3\sin(\theta_2+\theta_3) \\ &\quad + (d_4 + d_{4_{cte}})\cos(\theta_2+\theta_3)) \end{aligned} \tag{1}$$

$$\begin{aligned} y_{bt} &= \sin\theta_1(-a_1 + a_2\cos\theta_2 - a_3\sin(\theta_2+\theta_3) \\ &\quad + (d_4 + d_{4_{cte}})\cos(\theta_2+\theta_3)) \end{aligned} \tag{2}$$

$$\begin{aligned} z_{bt} &= d_1 + a_2\sin\theta_2 + a_3\cos(\theta_2+\theta_3) \\ &\quad + (d_4 + d_{4_{cte}})\sin(\theta_2+\theta_3) \end{aligned} \tag{3}$$

2) Grapple Kinematics and Dynamics: We model the grapple with two revolute joints, corresponding two links and point masses on the two links, to capture the mass distribution of the grapple components (see Fig. 2). Point $MC0$ is the center of mass (COM) of the first grapple part, and point MC is the COM of the next grapple parts, including rotator, claws, cylinder and piston, etc. Frame F_{C1}, whose origin is at the boom tip, rotates with the first crane link, i.e., through angle θ_1 about the z_0-axis. F_{G5} and F_{G6} are attached to the first and second grapple links, respectively. The Angle θ_5 represents the rotation between F_{C1} and F_{G5} about $-y$-axis, while θ_6 represents the rotation between F_{G5} and F_{G6} about x-axis, as shown in Fig. 2.



i	$Rot_z(\theta_i)$	$Trans_z(d_i)$	$Trans_x(a_i)$	$Rot_x(\alpha_i)$
1	θ_1	d_1	$-a_1$	$\pi/2$
2	θ_2	0	a_2	0
3	$\theta_3+\pi/2$	0	a_3	$\pi/2$
4	0	d_4+d_{4cte}	0	0

d_1(m)	a_1(m)	a_2(m)	a_3(m)	d_{4cte}(m)
1.43	0.15	4.35	0.43	2.39

Fig. 1: The forwarder crane and grapple. DH frames and tables are also shown. F_0 is the global frame.

We aim to find the cartesian positions of point masses m_0 and m in the global frame. First we consider m. Position of m relative to the boom tip expressed in F_{C1} is:

$$(x_{m/bt})_{F_{c1}} = l_3\sin\theta_5 \tag{4}$$

$$(y_{m/bt})_{F_{c1}} = l_2\sin\theta_6 \tag{5}$$

$$(z_{m/bt})_{F_{c1}} = -l_3\cos\theta_5 \tag{6}$$

where l_1 is the distance between the first and second passive joints, and l_2 is the distance between the second passive joint and point MC, and l_3 is equal to $l_1 + l_2\cos\theta_6$.

The global position of m expressed in F_0 is:

$$x_m = x_{bt} + (x_{m/bt})_{F_{c1}}\cos\theta_1 - (y_{m/bt})_{F_{c1}}\sin\theta_1 \tag{7}$$

$$y_m = y_{bt} + (x_{m/bt})_{F_{c1}}\sin\theta_1 + (y_{m/bt})_{F_{c1}}\cos\theta_1 \tag{8}$$

$$z_m = z_{bt} + (z_{m/bt})_{F_{c1}} \tag{9}$$

The global position of m_0 is found with the same procedure.

The dynamics equations governing the grapple's sway can be determined by following the Lagrangian approach to give:

$$\begin{aligned} -b_5\dot\theta_5 &= (m_0l_0 + ml_3)\big(\cos\theta_5\cos\theta_1\ddot x_{bt} + \cos\theta_5\sin\theta_1\ddot y_{bt} \\ &\quad + \sin\theta_5(\ddot z_{bt} + g)\big) + m_0l_0(l_0\ddot\theta_5 - l_0\sin\theta_5\cos\theta_5\dot\theta_1^2) \\ &\quad + ml_3\big(l_3\ddot\theta_5 - 2l_2\sin\theta_6\dot\theta_5\dot\theta_6 - l_2\cos\theta_5\sin\theta_6\ddot\theta_1 \\ &\quad - l_3\sin\theta_5\cos\theta_5\dot\theta_1^2 - 2l_2\cos\theta_5\cos\theta_6\dot\theta_1\dot\theta_6\big) \end{aligned} \tag{10}$$

$$\begin{aligned} -b_6\dot\theta_6 &= ml_2\big(\ddot x_{bt}(-\cos\theta_6\sin\theta_1 - \sin\theta_5\sin\theta_6\cos\theta_1) \\ &\quad + \ddot y_{bt}(\cos\theta_6\cos\theta_1 - \sin\theta_5\sin\theta_6\sin\theta_1) + l_2\ddot\theta_6 \\ &\quad + \cos\theta_5\sin\theta_6(\ddot z_{bt} + g) + \sin\theta_5(l_1\cos\theta_6 + l_2)\ddot\theta_1 \\ &\quad + \sin\theta_6(l_1 - l_3\cos^2\theta_5)\dot\theta_1^2 + l_3\sin\theta_6\dot\theta_5^2 \\ &\quad + 2l_3\cos\theta_5\cos\theta_6\dot\theta_1\dot\theta_5\big) \end{aligned} \tag{11}$$

where the damping torques at the joints are modeled as viscous friction with damping coefficients b_5 and b_6. The grapple's properties are given in Table I.

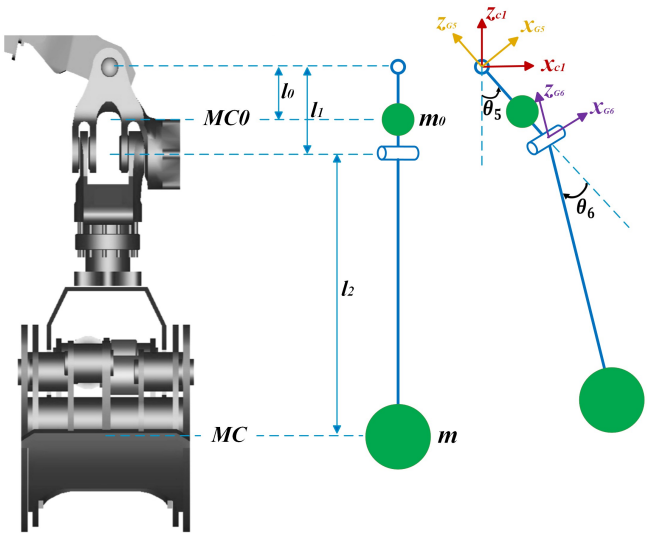


Fig. 2: The grapple model. θ_5 and θ_6 are sway angles.

TABLE I: The grapple's properties

l_0(m)	l_1(m)	l_2(m)	m_0(kg)	m(kg)	b_5(Nms/rad)	b_6(Nms/rad)
0.14	0.25	0.80	65	495	200	250

The accelerations $\ddot{\theta}_5$ and $\ddot{\theta}_6$ can be extracted from (10) and (11) as a function of other terms. In addition, $\ddot{x}_{bt}$, $\ddot{y}_{bt}$, $\ddot{z}_{bt}$ can be substituted with the second derivatives of (1) to (3). Although it increases the complexity of the sway dynamics equations, this formulation bypasses having to solve the inverse kinematics separately, i.e., it is solved automatically by solving DP, thereby lowering the computational cost, especially for a redundant crane.

III. Dynamic Programming for Trajectory Generation

Dynamic programming (DP) is an optimization algorithm based on Bellman's "principle of optimality", which states that an optimal sequence of decisions has the property that whatever the initial states and initial decisions are, the remaining decisions are optimal for the remaining problem. Therefore, a series of optimal values can be calculated recursively backward, starting from the final state.

A. Algorithm Description

The procedure presented here follows the approach of applying DP to a discrete piecewise linear system, based on [23], [24]. The general equation of such a system is:

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}_k\boldsymbol{x}_k + \boldsymbol{B}_k\boldsymbol{u}_k, \tag{12}$$

where $\boldsymbol{x}_k$ is the k-th state, and $\boldsymbol{u}_k$ is the k-th input.

Given an initial state $\boldsymbol{x}_1$, our goal is to find the optimal sequence of inputs minimizing the objective function shown in (13), which can be a general quadratic function.

$$\Gamma(\boldsymbol{x},\boldsymbol{u}) = \sum_{k=1}^{N}\Gamma_k(\boldsymbol{x}_k,\boldsymbol{u}_k) \tag{13}$$

$$\Gamma_k = \eta_k + \boldsymbol{x}_k^T\boldsymbol{y}_k + \boldsymbol{u}_k^T\boldsymbol{z}_k + \frac{1}{2}[\boldsymbol{x}_k^T\boldsymbol{Q}_k\boldsymbol{x}_k + 2\boldsymbol{x}_k^T\boldsymbol{R}_k\boldsymbol{u}_k + \boldsymbol{u}_k^T\boldsymbol{S}_k\boldsymbol{u}_k] \tag{14}$$

where $\eta_k, \boldsymbol{y}_k, z_k, \boldsymbol{Q}_k, \boldsymbol{R}_k, \boldsymbol{S}_k$ are time-varying coefficients. The optimal value function, defined in (15), is also quadratic in form, like (16), since the objective function was quadratic.

$$\Lambda_i = \min_{(u_i \dots u_N)} \sum_{k=i}^{N}\Gamma_k(\boldsymbol{x}_k,\boldsymbol{u}_k). \tag{15}$$

$$\Lambda_i(\boldsymbol{x}_i) = \zeta_i + \boldsymbol{x}_i^T\boldsymbol{v}_i + \frac{1}{2}\boldsymbol{x}_i^T\boldsymbol{W}_i\boldsymbol{x}_i, \tag{16}$$

where $\zeta_i, \mathbf{v}_i, \mathbf{W}_i$ are the coefficients. The backward recursive relation can be formed by employing the Bellman's principle:

$$\Lambda_i = \min_{u_i}(\Gamma_i + \Lambda_{i+1}). \tag{17}$$

Substituting (14) and (16) into (17) and using (12) leads to:

$$\zeta_i + \boldsymbol{x}_i^T\boldsymbol{v}_i + \frac{1}{2}\boldsymbol{x}_i^T\boldsymbol{W}_i\boldsymbol{x}_i = \min_{\boldsymbol{u}_i}\Bigg\{\zeta_{i+1} + \eta_i + \boldsymbol{x}_i^T\boldsymbol{h}_{4i} + \boldsymbol{u}_i^T\boldsymbol{h}_{5i} + \frac{1}{2}\Big[\boldsymbol{x}_i^T\boldsymbol{H}_{1i}\boldsymbol{x}_i + 2\boldsymbol{x}_i^T\boldsymbol{H}_{2i}\boldsymbol{u}_i + \boldsymbol{u}_i^T\boldsymbol{H}_{3i}\boldsymbol{u}_i\Big]\Bigg\} \tag{18}$$

where

$$\begin{aligned} &\boldsymbol{H}_{1i} = \boldsymbol{Q}_i + \boldsymbol{A}_i^T\boldsymbol{W}_{i+1}\boldsymbol{A}_i \quad , \quad \boldsymbol{H}_{2i} = \boldsymbol{R}_i + \boldsymbol{A}_i^T\boldsymbol{W}_{i+1}\boldsymbol{B}_i \\ &\boldsymbol{H}_{3i} = \boldsymbol{S}_i + \boldsymbol{B}_i^T\boldsymbol{W}_{i+1}\boldsymbol{B}_i \quad , \quad \boldsymbol{h}_{4i} = \boldsymbol{y}_i + \boldsymbol{A}_i^T\boldsymbol{v}_{i+1} \\ &\boldsymbol{h}_{5i} = \boldsymbol{z}_i + \boldsymbol{B}_i^T\boldsymbol{v}_{i+1} \end{aligned} \tag{19}$$

Differentiating the right-hand side of (18) with respect to $\boldsymbol{u}_i$ and equating it to zero gives the optimal input:

$$\boldsymbol{u}_i = -\boldsymbol{H}_{3i}^{-1}[\boldsymbol{H}_{2i}^T\boldsymbol{x}_i + \boldsymbol{h}_{5i}] \tag{20}$$

Substituting this $\boldsymbol{u}_i$ into (18) and equating terms of same degree in $\boldsymbol{x}_i$ yield the following set of recursive equations:

$$\begin{aligned} \zeta_i &= \zeta_{i+1} + \eta_i - \frac{1}{2}\boldsymbol{h}_{5i}^T\boldsymbol{H}_{3i}^{-1}\boldsymbol{h}_{5i} \\ \boldsymbol{v}_i &= \boldsymbol{h}_{4i} - \boldsymbol{H}_{2i}\boldsymbol{H}_{3i}^{-1}\boldsymbol{h}_{5i} \\ \boldsymbol{W}_i &= \boldsymbol{H}_{1i} - \boldsymbol{H}_{2i}\boldsymbol{H}_{3i}^{-1}\boldsymbol{H}_{2i}^T \end{aligned} \tag{21}$$

where the values at the terminal N-th stage to initialize the DP are given by:

$$\zeta_N = \eta_N \quad , \quad \boldsymbol{v}_N = \boldsymbol{y}_N \quad , \quad \boldsymbol{W}_N = \boldsymbol{Q}_N \tag{22}$$

1) Definition of States, Inputs, and Objective Function: We regard the position and velocity of the six joints as states, and the acceleration of the first four joints as inputs.

$$\boldsymbol{x} = [\theta_1\ \dot{\theta}_1\ \theta_2\ \dot{\theta}_2\ \theta_3\ \dot{\theta}_3\ d_4\ \dot{d}_4\ \theta_5\ \dot{\theta}_5\ \theta_6\ \dot{\theta}_6]^T \tag{23}$$

$$\boldsymbol{u} = [\ddot{\theta}_1\ \ddot{\theta}_2\ \ddot{\theta}_3\ \ddot{d}_4]^T \tag{24}$$

The desired final states, considering the anti-sway objective for the generated trajectory, and the initial states are determined in (25) and (26), respectively.

$$\boldsymbol{x}_f = [\theta_{1f}\ 0\ \theta_{2f}\ 0\ \theta_{3f}\ 0\ d_{4f}\ 0\ 0\ 0\ 0\ 0]^T \tag{25}$$

$$\boldsymbol{x}_i = [\theta_{1i}\ \ 0\ \theta_{2i}\ 0\ \theta_{3i}\ \ 0\ d_{4i}\ 0\ 0\ 0\ 0\ 0]^T \tag{26}$$

In non-rest-to-rest conditions, the last four states at (26) are non-zero.

We consider the following objective function:

$$\Gamma = \left\{ \sum_{k=1}^{N} \frac{1}{2} \boldsymbol{u}_k^T \boldsymbol{u}_k \right\} + \frac{1}{2} p [\boldsymbol{x}_N - \boldsymbol{x}_f]^T [\boldsymbol{x}_N - \boldsymbol{x}_f] \quad (27)$$

The first term of (27) enforces a minimum-effort condition, and the second term enforces the desired final condition via a penalty weight p. Comparing (27) and (13), (14) results in:

$$\eta_N = \frac{1}{2} p \boldsymbol{x}_f^T \boldsymbol{x}_f \ , \ \boldsymbol{y}_N = -p \boldsymbol{x}_f \ , \ \boldsymbol{Q}_N = p \boldsymbol{I}_{12} \ , \ \boldsymbol{S}_k = \boldsymbol{I}_4 \quad (28)$$

Other coefficients of (13), (14) are equal to zero.

B. Linearization Approaches

The aforementioned equations hold when the discrete system is piecewise linear. To linearize the sway dynamics to obtain the linear state and input matrices $\boldsymbol{A}_k$ and $\boldsymbol{B}_k$ in (12), we use the linearization about the nominal trajectory (LNT) approach and show its superiority in the DP framework as compared to linearization about the current states and inputs (LCSI) approach, used in previous works.

1) Linearization about Nominal Trajectory: The general state-space representation of the system is given by:

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, t) \quad (29)$$

Linearizing (29) about a nominal trajectory results in:

$$\dot{\boldsymbol{x}} - \dot{\bar{\boldsymbol{x}}} = \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \right|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}} (\boldsymbol{x} - \bar{\boldsymbol{x}}) + \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \right|_{\bar{\boldsymbol{x}}, \bar{\boldsymbol{u}}} (\boldsymbol{u} - \bar{\boldsymbol{u}}) \quad (30)$$

where $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}$ are the nominal states and inputs.

To discretize (30), the Euler method is used:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + h \dot{\boldsymbol{x}} \quad (31)$$

The discrete linearized system, using (30) and (31), is:

$$\begin{aligned} \boldsymbol{x}_{k+1} - \bar{\boldsymbol{x}}_{k+1} = (\boldsymbol{I} + h \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \right|_{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k}) (\boldsymbol{x}_k - \bar{\boldsymbol{x}}_k) \\ + h \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \right|_{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k} (\boldsymbol{u}_k - \bar{\boldsymbol{u}}_k) \end{aligned} \quad (32)$$

$$\boldsymbol{x}_{k+1}^* = \boldsymbol{A}_k \boldsymbol{x}_k^* + \boldsymbol{B}_k \boldsymbol{u}_k^* \quad (33)$$

where $\bar{\boldsymbol{x}}_k$ and $\bar{\boldsymbol{u}}_k$ are the nominal states and inputs at t_k, and:

$$\boldsymbol{x}_k^* = \boldsymbol{x}_k - \bar{\boldsymbol{x}}_k \quad , \quad \boldsymbol{u}_k^* = \boldsymbol{u}_k - \bar{\boldsymbol{u}}_k \quad (34)$$

$$\boldsymbol{A}_k = \boldsymbol{I} + h \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \right|_{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k} , \quad \boldsymbol{B}_k = h \left. \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}} \right|_{\bar{\boldsymbol{x}}_k, \bar{\boldsymbol{u}}_k} \quad (35)$$

DP algorithm is applied on (33), hence, $\boldsymbol{u}_k^*$ will be found. Then, we can easily calculate the optimal inputs for the real system by $\boldsymbol{u}_k = \boldsymbol{u}_k^* + \bar{\boldsymbol{u}}_k$

The initial states and desired final states for our DP are:

$$\boldsymbol{x}_i^* = \boldsymbol{x}_i - \bar{\boldsymbol{x}}_1 \quad (36)$$

$$\boldsymbol{x}_f^* = \boldsymbol{x}_f - \bar{\boldsymbol{x}}_N \quad (37)$$

where $\bar{\boldsymbol{x}}_1$ and $\bar{\boldsymbol{x}}_N$ are obtained by numerical simulation of system using the nominal trajectory first.

TABLE II: Conditions of Maneuvers for each subsection

	time (s)	θ_1 (deg)	θ_2 (deg)	θ_3 (deg)	d_4 (m)
A	t_i = 0	0	0	0	0
	t_f = 3.5	-30	-30	35	1.2
B	t_i = 0	0	55	-113	0.5
	t_f = 3.5	45	35	-93	1.5
C	t_i = 0	45	35	-93	1.5
	t_f = 3.5	0	55	-113	0.5

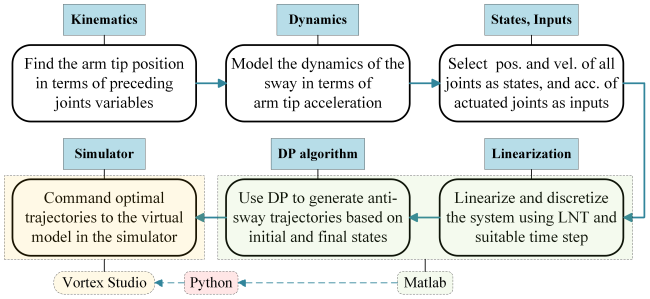


Fig. 3: Schematic diagram of our proposed methodology.

2) Linearization about Current States and Inputs: In this method, derivatives in (35) are evaluated at the current state and input values $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$; however, $\boldsymbol{u}_k$ is unknown prior to completing the DP recursion. Therefore, an initial trajectory estimate is required for the first optimization pass, followed by several passes (DP execution, and $\boldsymbol{A}_k$, $\boldsymbol{B}_k$ evaluations) needed to reach an acceptable level of convergence. The details of this approach can be found in [17].

IV. Simulation Results and Discussion

The DP algorithm was implemented in Matlab, and the numerical simulations required for the linearization in the DP framework were carried out with Matlab ode45 solver. Then, the DP trajectory solutions were commanded to the virtual model of the forwarder, created in a high-fidelity multibody-dynamics simulator, *Vortex Studio* [25], with Python scripts for interface to Matlab, to validate the results. The forwarder model we used is based on the Tigercat 1075C forwarder and models the realistic grapple dynamics. The nominal trajectory used for LNT and the initial trajectory for LCSI are 5^{th} order polynomials (5OP) with zero initial and final velocities and accelerations. Three test scenarios (A, B, C) were considered with initial and final conditions of the corresponding P2P maneuvers summarized in Table II. A schematic diagram representing our proposed methodology, which employs DP and LNT approaches, is shown in Fig. 3.

A. Effect of Linearization Scheme

Fig. 4 shows the sway angles for maneuver A (see Table II) obtained with the two linearization schemes, LCSI and LNT. The integration time step was $0.001s$, and all simulation results for this case were obtained with Matlab ode45 solver. As shown, the LCSI results, even after four passes of the DP algorithm, are inferior to the LNT results. The DP trajectories obtained with LNT suppress the residual sway to less than 5% of the sway resulting when using 5OP trajectories; on the other hand, the trajectories for the LCSI approach show only

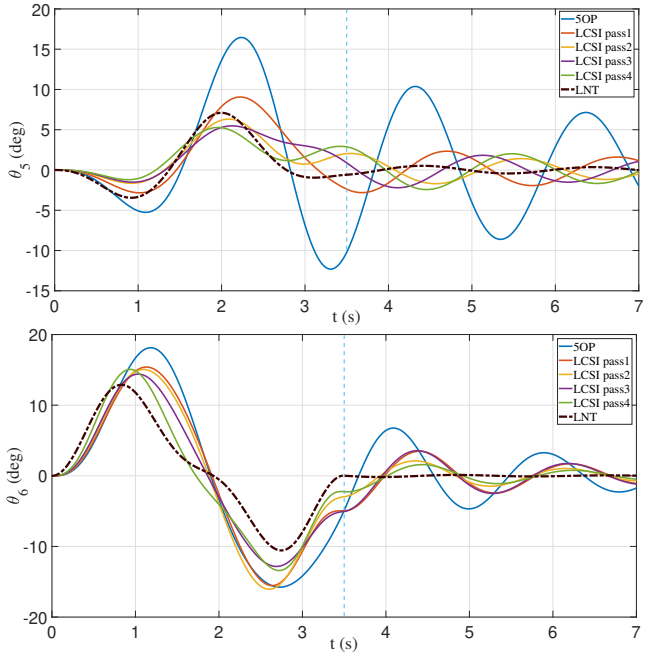


Fig. 4: The effect of the linearization scheme on sway angles θ_5 and θ_6 in maneuver A. The dashed vertical line indicated the end time of the maneuver (t_f).

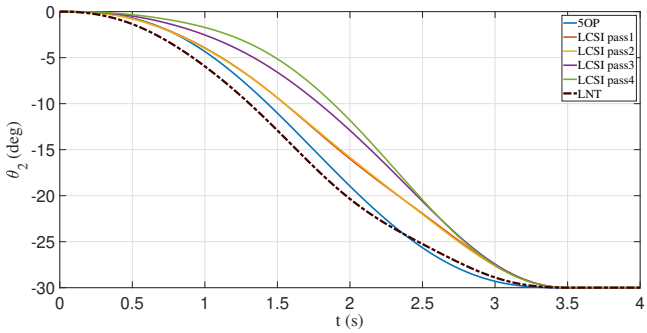


Fig. 5: The effect of the linearization scheme on the DP trajectories for θ_2 in maneuver A.

a 30% reduction in sway after the fourth pass. In addition to higher computational cost and inferior results, using LCSI requires a careful selection of p (penalty weight) in the DP framework to obtain adequate convergence with successive passes [17]. Fig. 5, representing the second joint trajectories, as an example, shows the differences between the optimal trajectories obtained when using LCSI and LNT. Therefore, we employed the LNT approach for the rest of our work.

B. Results for Virtual Forwarder Model

The maneuver considered here involves repositioning the boom tip from a location above the forwarder basket to a log pile on the side of the forwarder, which is the most common P2P maneuver of the forwarder crane. Its conditions are noted in Table II (maneuver B). One important point in working with real cranes is the time step by which we can command them. This time step is $0.0167s$ (equivalent to $60Hz$) for our virtual forwarder model. Here, we solved the DP algorithm with the time step of $0.0001s$, and re-

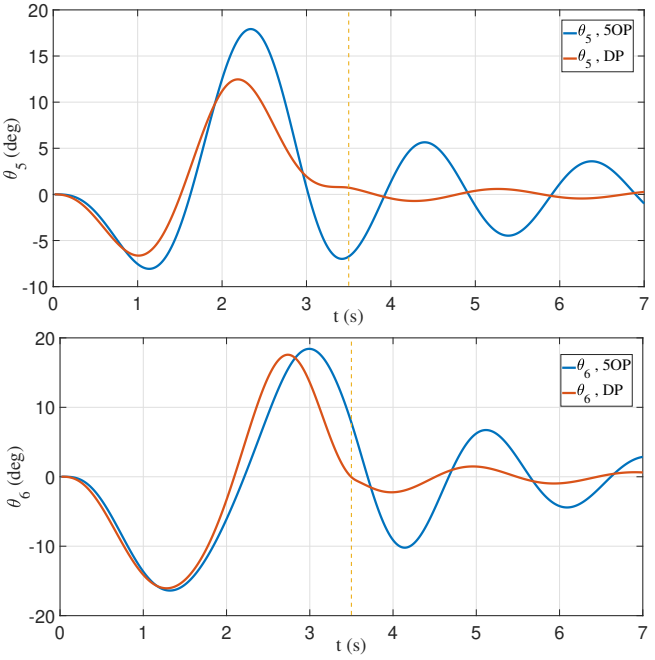


Fig. 6: Sway angles θ_5 and θ_5 responses of the virtual forwarder model with 5OP and DP commanded trajectories in maneuver B. The dashed vertical line indicated the end time of the maneuver (t_f).

sampled the DP trajectories' points at a $0.0167s$ period for compatibility with the virtual model. Therefore, in all of the following results, $0.0167s$ was used as the simulation time step. Fig. 6 depicts the grapple sway angles computed by using 5OP versus DP trajectories and Fig. 7 shows the corresponding trajectories for the actuated joints. As evident, the DP generated trajectories are smooth and effective at damping the residual sway, with the maximum values of θ_5 and θ_6 of $0.6°$ and $2.0°$, respectively, after t_f, which are suppressed to be less than 8% and 19% of those numbers while using 5OP trajectories. By increasing the sampling rate on the hardware or of the virtual forwarder model, these numbers can be further improved.

C. Non-Rest-To-Rest Conditions

Initial sway conditions cannot be always considered to be zero in real-world applications. Given the initial sway conditions, DP can generate non-rest-to-rest trajectories at no additional cost by changing the last four initial states in (26). We considered non-small initial sway conditions, mentioned in Table III, along with other conditions of maneuver C in Table II. We ran the DP algorithm with the time step equal to $0.0001s$ and resampled the DP trajectories' points at a $0.0167s$ period to apply them to the virtual model. The results, illustrated in Fig. 8, show that DP trajectories could attenuate the residual sway effectively even in the presence of the initial sway. The maximum values of the residual sway are $0.2°$ and $1.6°$, which are damped to be less than 1% and 23% of those numbers when 5OP trajectories were used. DP and 5OP trajectories commanded to the third joint, as an example, are shown in Fig. 9.

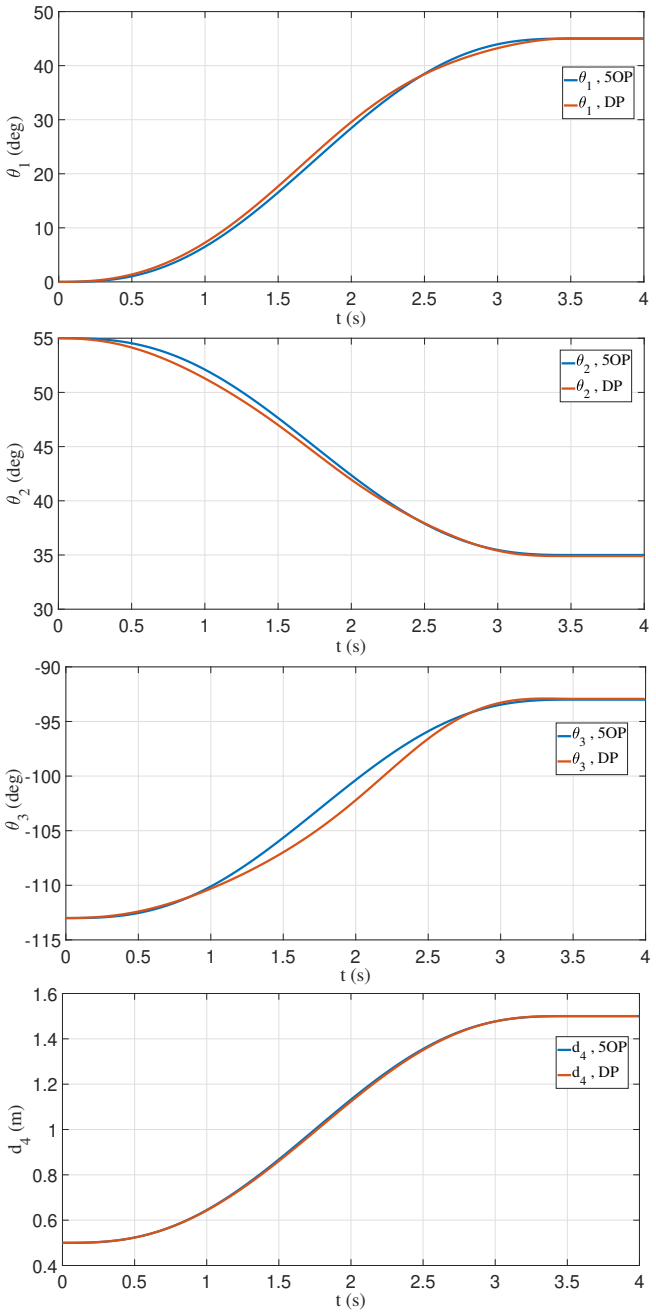


Fig. 7: 5OP and DP commanded trajectories to the first four joints in the virtual model for maneuver B.

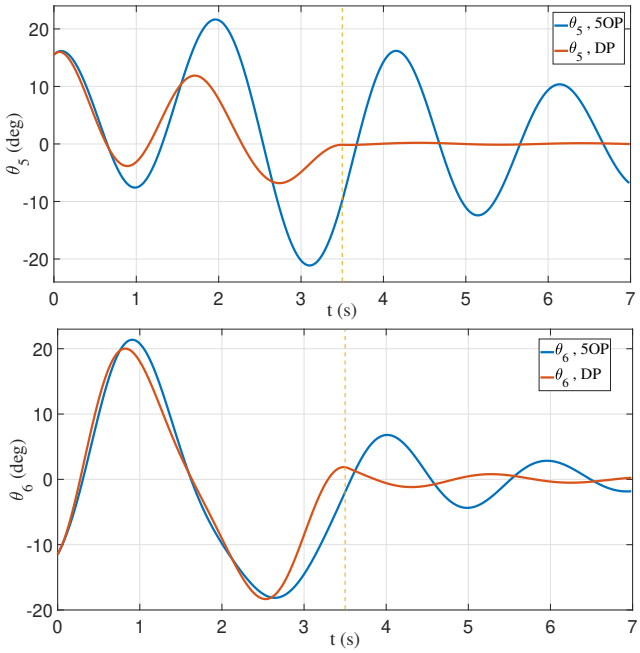


Fig. 8: Sway angles θ_5 and θ_5 responses of the virtual forwarder model with 5OP and DP commanded trajectories in the non-rest-to-rest conditions of maneuver C. The dashed vertical line indicated the end time of the maneuver (t_f).

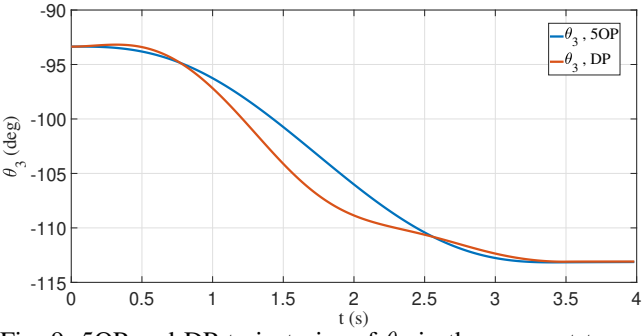


Fig. 9: 5OP and DP trajectories of θ_3 in the non-rest-to-rest conditions of maneuver C in the virtual model.

TABLE III: Initial Sway Conditions of Maneuver C

θ_{5i} (deg)	$\dot{\theta}_{5i}$ (deg/s)	θ_{6i} (deg)	$\dot{\theta}_{6i}$ (deg/s)
15.5	18.5	-11.5	21

V. Conclusion and Future Work

We proposed a general methodology for solving the sway problem of passive end-effectors attached to serial robotic manipulators. The method is based on the 3D sway dynamics, formulated in terms of manipulator actuated joint variables. Dynamic programming, combined with the linearization about a nominal trajectory approach, are employed to generate anti-sway trajectories for each actuated joint. We evaluated the methodology on the forestry forwarder, where the sway problem is detrimental, by using its virtual model developed in a high-fidelity multibody-dynamics simulator. The results illustrate the effectiveness of our method at damping the grapple's residual sway, specifically, a reduction of 90% on average, compared to using quintic polynomial trajectories. We also investigated how the choice of the linearization approach affects the computational cost, as well as the trajectory produced by the DP solution. In addition, we showed that our methodology can be used well for both rest-to-rest and non-rest-to-rest maneuver conditions. Our next steps are to define an optimization framework for finding the minimum possible t_f (maneuver time) while ensuring sway damping, and also to integrate the trajectory generator with a path planner.

Acknowledgments

This work was supported by the National Sciences and Engineering Research Council (NSERC) Canadian Robotics Network (NCRN), and MITACS Accelerate Award.

REFERENCES

[1] Y. P. Springer, D. L. Lucas, L. J. Castrodale, and J. B. McLaughlin, "Work-related injuries in the alaska logging industry, 1991-2014," *American journal of industrial medicine*, vol. 61, no. 1, pp. 32–41, 2018.

[2] M.-D. Iftime, A.-E. Dumitrascu, D.-I. Dumitrascu, and V. D. Ciobanu, "An investigation on major physical hazard exposures and health effects of forestry vehicle operators performing wood logging processes," *International Journal of Industrial Ergonomics*, vol. 80, p. 103041, 2020.

[3] R. Malinovski, J. Malinovski, L. Nutto, and N. Sanches, *Safety and Training in Harvesting*, 01 2016, pp. 2521–2560.

[4] J. Dvořák, Z. Malkovskỳ, and J. Mack, "Influence of human factor on the time of work stages of harvesters and crane-equipped forwarders," *Journal of Forest Science*, vol. 54, no. 1, pp. 24–30, 2008.

[5] S. Joshi, S. Kumra, and F. Sahin, "Robotic grasping using deep reinforcement learning," in *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2020, pp. 1461–1466.

[6] S. Fodor, C. Vázquez, and L. Freidovich, "Automation of slewing motions for forestry cranes," in *2015 15th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2015, pp. 796–801.

[7] P. Mustalahti, J. Koivumäki, and J. Mattila, "Stability-guaranteed anti-sway controller design for a redundant articulated hydraulic manipulator in the vertical plane," in *Fluid Power Systems Technology*, vol. 58332. American Society of Mechanical Engineers, 2017, p. V001T01A031.

[8] J. Kalmari, J. Backman, and A. Visala, "Nonlinear model predictive control of hydraulic forestry crane with automatic sway damping," *Computers and Electronics in Agriculture*, vol. 109, pp. 36–45, 2014.

[9] J. Kalmari, H. Hyyti, and A. Visala, "Sway estimation using inertial measurement units for cranes with a rotating tool," *IFAC Proceedings Volumes*, vol. 46, no. 10, pp. 274–279, 2013.

[10] N. Yanai, M. Yamamoto, and A. Mohri, "Feedback control for wire-suspended mechanism with exact linearization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3. IEEE, 2002, pp. 2213–2218.

[11] Y. Wu, N. Sun, H. Chen, and Y. Fang, "Adaptive output feedback control for 5-dof varying-cable-length tower cranes with cargo mass estimation," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2453–2464, 2020.

[12] M. J. Agostini, G. G. Parker, H. Schaub, K. Groom, and R. D. Robinett, "Generating swing-suppressed maneuvers for crane systems with rate saturation," *IEEE Transactions on Control Systems Technology*, vol. 11, no. 4, pp. 471–481, 2003.

[13] R. E. Samin, Z. Mohamed, J. Jalani, and R. Ghazali, "Input shaping techniques for anti-sway control of a 3-dof rotary crane system," in *2013 1st International Conference on Artificial Intelligence, Modelling and Simulation*. IEEE, 2013, pp. 184–189.

[14] M. Maghsoudi, L. Ramli, S. Sudin, Z. Mohamed, A. Husain, and H. Wahid, "Improved unity magnitude input shaping scheme for sway control of an underactuated 3d overhead crane with hoisting," *Mechanical systems and signal processing*, vol. 123, pp. 466–482, 2019.

[15] T. M. Nguyen, T. H. Do, and M. D. Duong, "Anti-sway and position control for double-pendulum crane using is-adrc controller," *Measurement, Control, and Automation*, vol. 3, no. 1, pp. 3–9, 2022.

[16] D. Zameroski, G. Starr, J. Wood, and R. Lumia, "Swing-free trajectory generation for dual cooperative manipulators using dynamic programming," in *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.* IEEE, 2006, pp. 1997–2003.

[17] ——, "Rapid swing-free transport of nonlinear payloads using dynamic programming," *Journal of Dynamic Systems, Measurement, and Control*, vol. 130, no. 4, 2008.

[18] I. Palunko, R. Fierro, and P. Cruz, "Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 2691–2697.

[19] G. Starr, J. Wood, and R. Lumia, "Rapid transport of suspended payloads," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE, 2005, pp. 1394–1399.

[20] S. Fielding and M. Nahon, "Input shaped trajectory generation and controller design for a quadrotor-slung load system," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2019, pp. 162–170.

[21] A. Wahrburg, J. Jurvanen, M. Niemelä, and M. Holmberg, "Input shaping for non-zero initial conditions and arbitrary input signals with an application to overhead crane control," in *2022 IEEE 17th International Conference on Advanced Motion Control (AMC)*. IEEE, 2022, pp. 36–41.

[22] M. W. Spong, S. Hutchinson, M. Vidyasagar *et al.*, *Robot modeling and control*. Wiley New York, 2006, vol. 3.

[23] R. D. Robinett III, D. G. Wilson, G. R. Eisler, and J. E. Hurtado, *Applied dynamic programming for optimization of dynamical systems*. SIAM, 2005.

[24] D. Bertsekas, *Dynamic programming and optimal control: Volume I*. Athena scientific, 2012, vol. 1.

[25] "CM-Labs, Vortex-Studio." https://www.cm-labs.com/vortex-studio/software/modular-software-architecture-simulation/editor/.