Investigations into Exploiting the Full Capabilities of a Series-Parallel Hybrid Humanoid using Whole Body Trajectory Optimization

Melya Boukheddimi[1*], Rohit Kumar[1*], Shivesh Kumar[1], Justin Carpentier[2,3], and Frank Kirchner[1,4]

***Abstract*— Trajectory optimization methods have become ubiquitous for the motion planning and control of underactuated robots for e.g., quadrupeds, humanoids etc. While they have been extensively used in the case of serial or tree type robots, they are seldomly used for planning and control of robots with closed loops. Series-parallel hybrid topology is quite commonly used in the design of humanoid robots, but they are often neglected during trajectory optimization and the movements are computed for a serial abstraction of the system and then the solution is mapped to the actuator coordinates. As a consequence, the full capability of the robot cannot be exploited. This paper presents a case study of trajectory optimization for series-parallel hybrid robot by taking into account all the holonomic constraints imposed by the closed kinematic loops present in the system. We demonstrate the advantages of this consideration with a weightlifting task on RH5 Manus humanoid in both simulation and experiments.**

I. Introduction

The recent developments in robotics have seen a large adaptation of closed-loop mechanisms in various robots like exoskeletons [1], [2], multi-legged robots [3], humanoid robots [4], [5] etc. These parallel mechanisms provide higher stiffness, high payload capacities, higher precision, etc. Series-parallel hybrid robots can be defined as the combination of serial chain and parallel mechanisms which can bring together the advantages of both topologies. An extensive survey on these is available in [6]. They are often combined to closely mimic the human and animal capabilities which require high stiffness, optimum mass and inertia distribution properties, etc. While there are many advantages, these series-parallel hybrid robots also inherit the kinematic complexities of both serial and parallel architectures. Currently, the trajectory optimization approaches are very popular and powerful methods for motion planning. They can help in generating various complex movements for multi-body systems [7]. Most of the trajectory optimization problems are usually based on tree type systems as they are easier to model and control. Many of these solvers allow modeling of external kinematic constraints acting on the robot for e.g. humanoid legs while standing or in multi-contact scenarios [8], [9], [10]. Parallel robots have the advantages of high accuracy and rigidity with a large payload, however they involve loop closure constraints that are difficult to model and control. The optimization process is usually difficult or time-consuming, which makes it impossible to use in real-time control [11]. Some researchers have tried to control these systems using trajectory optimization approaches such as [12] where a cascade controller was proposed to control a pneumatic driven parallel robotic platform with optimal parameter tuning. However the intended tasks are industrial tasks, which means that they are repetitive and only involve kinematics. In [13], the authors proposed an optimal controller based on the firefly algorithm to generate optimal trajectories for a hydraulically parallel robot. In contrast, a complex series-parallel hybrid robot is difficult to solve since it is a combination of many parallel robots. In addition to optimal control formulation, resolving loop closure constraints is a burden on the optimization. When it comes to direct methods, [14] has proposed DIRCON a direct collocation algorithm which can effectively deal with kinematic constraints in both trajectory generation and stabilization steps with third-order integration accuracy. However, the method was applied to deal only with external kinematic constraints arising from contacts. Various shooting methods have also been proposed in the literature for whole body trajectory optimization [15] which can deal with kinematic constraints like contacts but have not been studied for kinematic loops within the robot. Most previous studies using shooting methods considered a serial abstraction of the series-parallel hybrid robot and the results were mapped to the closed-loop mechanisms present in the system using an additional mapping [5], [16], [17].

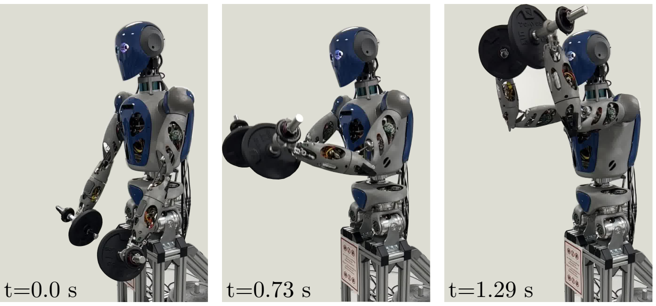


Fig. 1. Screenshot of the RH5 Manus robot performing a 15 kg lifting motion. The motion is generated using the constrained series-parallel model of the robot in the trajectory optimization process.

[*] Both authors contributed equally to the paper.

This work was supported by the Humane-AI (FKZ H2020-ICT-2019-3 #952026) project funded through European Unions Horizon 2020 research and innovation programme and additionally through the M-RoCK (FKZ 01IW21002), KiMMI-SF (FKZ 50RA2021, 50RA2022) and VeryHuman (FKZ 01IW20004) projects funded by the German Aerospace Center (DLR) with federal funds from the Federal Ministry of Education and Research (BMBF) and Federal Ministry of Economic Affairs and Energy (BMWi) respectively. The authors also acknowledge the help of Mahdi Javadi with multimedia creation.

[1] Robotics Innovation Center, DFKI GmbH, 28359 Bremen, Germany.
[2] INRIA Paris, 75012 France.
[3] Département d'informatique, ENS Paris, France.
[4] AG Robotik University of Bremen, 28359 Bremen, Germany.
Corresponding author: melya.boukheddimi@dfki.de

This process has the following disadvantages:

- Box constraints used to model the physical limitations of the mechanisms either overestimate or underestimate the effective workspace of the robot (see Section III).
- Parallel mechanisms may be subject to singularities that are not taken into account in the optimization problem while working with serial models.
- The optimization formulation is not accurate since it does not take into account the full dynamics of the closed-loop mechanisms of the system (see [18] for a case study on the involved trade-offs).

Contribution: In this paper, we propose a first case study on resolving all the loop-closure constraints of the robot within the trajectory optimization process. To this end, we use the open-source software Pinocchio with its recently introduced proximal formulation of the constrained dynamics [19]. This approach allows us to converge to an optimal solution according to the least squares principle, even in the context of singularities. Among the optimization methods available in the literature, the differential dynamics programming (DDP) approach was used to generate optimal trajectories with respect to the constrained dynamics. We consider the weight lifting task with 15 Kg in two arms for the RH5 Manus humanoid and demonstrate that by planning the trajectories directly in the actuation space, we can exploit the full capabilities of the robot which is not possible when working with a serial abstraction of the robot model. Results are shown in simulation as well as experiments on the real robot. This work is significant for humanoid robots based on electric actuation where one must seek to push the robot to its limits to achieve human like agility.

Organization: Section II presents the mathematical preliminaries for whole body trajectory optimization with kinematic constraints. Section III presents the design of the study on RH5 Manus humanoid platform for weight lifting task. Section IV presents the results and discussion and Section V concludes the paper and highlight our future work.

II. Mathematical Background

Series-parallel hybrid robots are subjected to large number of holonomic constraints. These constraints can act *externally* on the system, e.g. multiple contacts with the environment or *internally*, e.g. loop closure constraints for closed-loop mechanisms present within the system. This section presents the mathematical preliminaries of constrained dynamics formulation and the corresponding trajectory optimization problem.

A. Constrained Multi-Body Dynamics

The unconstrained multi-body dynamics of the robot is written in the following canonical form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}+\mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) = \tau + \sum_{k=1}^{K}\mathbf{J}_k^{\intercal}(\mathbf{q})\phi_k \tag{1}$$

where, $\mathbf{q},\dot{\mathbf{q}},\ddot{\mathbf{q}}$ corresponds to the vector of generalized positions, velocities and accelerations, $\mathbf{M}(\mathbf{q})$ is the generalized mass-inertia matrix, $\mathbf{b}(\mathbf{q},\dot{\mathbf{q}})$ is the bias force vector which includes Coriolis-Centrifugal and gravity forces and τ is the vector of generalized torques. K accounts for the additional kinematic constraints on the robot, such as external contact with the environment and internal kinematic loops. At position level, the kinematic constraints could be implicitly written as $\boldsymbol{f}_c(\mathbf{q}) = 0$. However, since we are solving the dynamics in the acceleration space, it is common to use the second derivatives of this constraint in our problem:

$$\mathbf{J}_k(\mathbf{q})\ddot{\mathbf{q}}+\dot{\mathbf{J}}_k(\mathbf{q})\dot{\mathbf{q}} = \boldsymbol{a}_c^* \tag{2}$$

$\mathbf{J}_k(\mathbf{q})$ is the Jacobian matrix corresponding to the k^{th} application of constraint on the robot. $\phi_k = [\boldsymbol{\lambda}_k \quad \eta_k]^T$ is the vector of the dual external forces ($\boldsymbol{\lambda}_k$) and the torques (η_k) that corresponds to the k^{th} constraints. $\boldsymbol{a}_c^*$ is the desired acceleration with corrective terms for constraint satisfaction.

1) Kinematic loop-closure as a rigid Constraints: Under kinematic constraints (i.e., kinematic loop closure), the dynamics of the robot is subject to the constrained equations of motion presented in (1) and (2), and this can be written as an optimization problem under equality constraints. The solution of the associated Lagrangian of this constrained dynamics (see [19] for full expansion) is then given by:

$$\begin{bmatrix} \mathbf{0} & \mathbf{J}_k \\ \mathbf{J}_k^{\intercal} & \mathbf{M} \end{bmatrix} \begin{bmatrix} -\boldsymbol{\phi_k} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} -\dot{\mathbf{J}}_k(\mathbf{q})\dot{\mathbf{q}} + \boldsymbol{a}_c^* \\ \tau - \mathbf{b}(\mathbf{q},\dot{\mathbf{q}}) \end{bmatrix}, \tag{3}$$

Multiple formulations have been proposed to solve this problem, for example using Schur's components of the cholesky-factorized constrained dynamics matrix [15] [20]. We use the recently proposed constrained dynamics algorithm in Pinocchio [19]. It reformulates the problem in a proximal way and provides a sparse way to efficiently handle this problem and its derivatives. We use the Pinocchio C++ Library for our formulations here, since it provides us with an efficient implementation of the unconstrained as well as constrained dynamics along with their analytical derivatives for help in optimization [19] [21][22].

B. Trajectory Optimization Formulation

The discrete time optimal control problem can be written as follows:

$$\min_{\boldsymbol{x},\boldsymbol{u}} l_N(\boldsymbol{x}_N) + \sum_{t=0}^{N-1} l(\boldsymbol{x}_t,\boldsymbol{\tau}_t)dt \tag{4a}$$

$$s.t. \quad \boldsymbol{x}_0 = \boldsymbol{f}_0, \tag{4b}$$

$$\forall i \in \{0...N-1\}, \;\; \boldsymbol{x}_{i+1} = \boldsymbol{f}_t(\boldsymbol{x}_i,\boldsymbol{\tau}_i) \tag{4c}$$

- $\boldsymbol{x} = (\mathbf{q},\dot{\mathbf{q}})$: robot state,
- $\boldsymbol{\tau} \in \mathbb{R}^{n_u}$: actuator effort,
- N : nodes number for the discretized trajectory,
- l_N : terminal cost model, applied on the last node of the trajectory,
- l : running cost model, applied on all remaining nodes,
- $\boldsymbol{f}_0$: the initial state of the problem $(\mathbf{q}_0,\dot{\mathbf{q}}_0)$,
- $\boldsymbol{f}_t$: the discretization of the robot dynamics (3).

Optimal trajectories are computed using the Box Feasibility DDP (BoxFDDP) solver proposed by the open-source C++ library Crocoddyl [15]. The Feasibility DDP [23] enables us to overcome the numerical limitations of the original single shooting DDP formulation [24]. The Box-FDDP [25] gives us the ability to reason about the actuation limits of the robot.

III. Experimental Design

This section presents the experimental design where we consider the fixed base RH5 Manus [16], a series-parallel hybrid robot with multiple kinematic loops for a weight lifting task using whole body trajectory optimization. Firstly, it describes the complete hybrid upper body of the robot with closed-loop mechanisms identifying the theoretical limits of the full robot and its tree abstraction. Next, it presents optimal control formulation for weight lifting with tree and full hybrid robot models.

A. Closed-loop mechanisms in RH5 Manus robot

The fixed base model of RH5 Manus robot consists of a total 61 spanning tree joints ($n = 61$). Among these, there are 20 independent joints ($m = 20$) and 20 active joints ($p = 20$). The topological graph of the robot can be seen in Fig. 2. All the independent joints are shown as green edges and the actuated joints are shown as red edges. Remaining spanning tree joints are passive in nature. The cut joints for loop closure is denoted by dotted lines. All closed-loop mechanisms in the robot are shown in blue boxes.

The series-parallel hybrid system in Fig. 2 can be represented as a tree type composition of 10 submechanisms. There are 5 serial chain submechanisms and 5 closed-loop submechanism. The first closed-loop submechanism connected to the root of the graph is a multi closed-loop torso mechanism of type 2S$\underline{P}$U+1U [26] and is actuated by two prismatic actuators (Joints 5 and 8), each on the left and right of the submechanism. Pitch and roll movements denotes the independent coordinates (Joints 1 and 2) of the submechanism. Each cut joint is a spherical joint and imposes 3D translation constraint in the submechanism.

The second closed-loop submechanism present in both arms of the robot is a planar closed-loop elbow mechanism of type RR$\underline{P}$R and is actuated by a prismatic actuator (joints 15 and 38 in right and left elbows). The elbow rotation is chosen as the independent coordinate (joints 13 and 36 in right and left elbows). In this submechanism, the cut joint is a revolute joint that impose planar translation constraints.

The third closed-loop submechanism in both arms is complex multi-loop closure wrist mechanism of type 2SU[RR$\underline{P}$R]+1U [27] and is actuated by two prismatic actuators (joints 28, 31, 51, and 54). Wrist pitch and yaw movements represents the independent coordinates (joints 17, 18, 40, and 41). There are multiple cut joints in the mechanism. One of the cut joint on both sides of the submechanism is a spherical joint that imposes 3D translation constraint. Other cut joint is a revolute joint on both sides imposing the planar translation constraints.

B. Box constraints

In an optimization problem, the physical limits of the robot's joints are generally modeled as box constraints [28], [25]. However, this is true for serial or tree type systems but is not necessarily a good argument for closed-loop mechanism [29]. In closed-loop mechanisms, the configuration space can be different from actuation space. When independent joints are considered, the physical limits of the actuators are masked and hence not exploited. There can be various configurations which are feasible in actuation space of the closed-loop mechanism but cannot be exploited while considering box constraints. The configuration analysis in independent joints space and actuation space of the closed-loop mechanisms in RH5 Manus have already been studied [30], [31], [27].

1) Torso: As an example, consider the torso submechanism in RH5 Manus robot in Fig. 3(a). The full admissible configuration space in the actuation coordinates is shown in Fig. 3(b). The box constraints are shown by blue lines in the plot as the actuator position limits. When this mechanism is modeled using independent joints, the configuration space can be seen in Fig. 3(c). The black curves represents the effective actuator limits, mapped from actuation space to independent joint space. However, due to box constraints, only green region can be reached acting as conservative limits. The full capabilities of the robot cannot be exploited.

2) Elbow: The same behavior can be observed on velocity and torque level for the elbow mechanism shown in Fig. 4(a). The maximum velocity of the linear actuator depends on the joint configuration, shown in Fig. 4(b). The box constraint provides the conservative limit for the maximum velocity, shown with dotted line. Similar conservative limits can be observed in Fig. 4(c) for maximum torque.

3) Wrist: Fig. 5(a) shows the wrist closed-loop mechanism in the robots and gives the same insight. The red lines act as box constraint for the actuation configuration space in Fig. 5(b). When the actuation space is mapped to independent joints configuration space, the box constraints act as conservative limits (shown with white dotted lines in Fig. 5(c)).

4) Overall ROM: The complete range of motion (ROM) for the linear actuators in the closed-loop mechanisms is summarized in Table I. The maximum force limits of the linear actuators reported here are lower than the ones reported in [16] in order to respect current limitations of the robot electronics. When box constraints on the actuators are mapped to independent joint space, the limits are configuration dependent and hence can take any value between maximum lower and maximum upper limits, summarized in Table II.

TABLE I
ROM of linear actuators used in RH5 Manus

Actuator	ROM (mm)	Max. force (N)	Max. vel. (mm/s)
Wrist	$[113, 178]$	612	200
Elbow	$[173.8, 295.42]$	921	266
Torso	$[195.0, 282.8]$	910	291

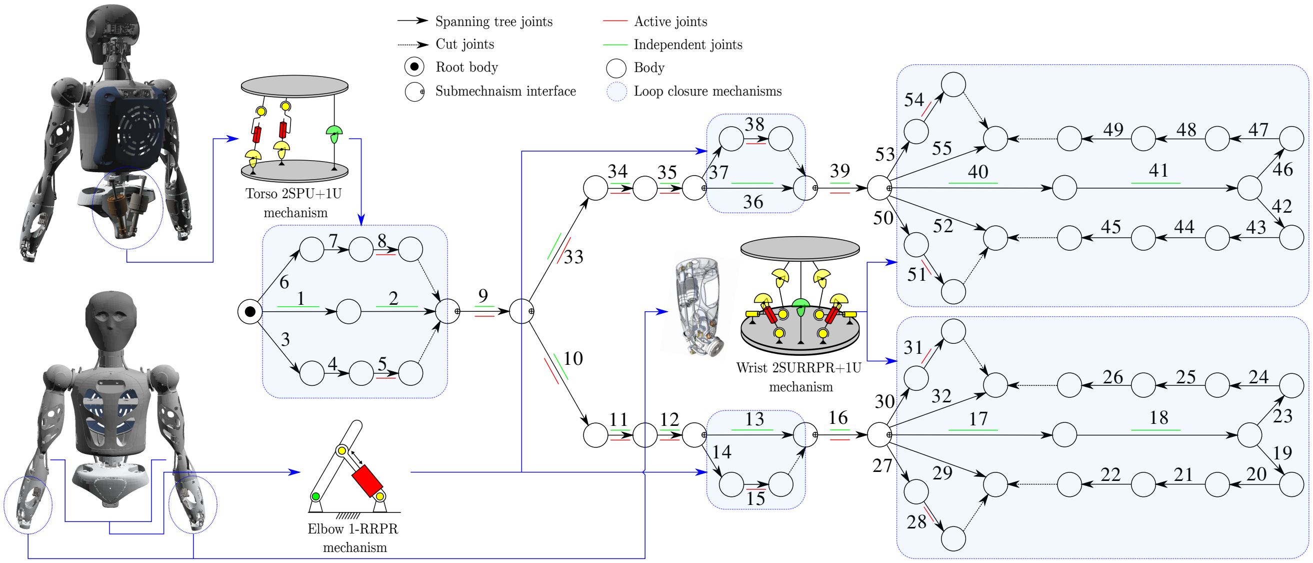
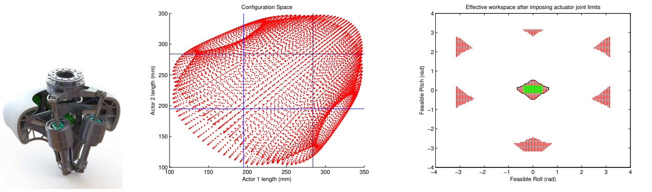
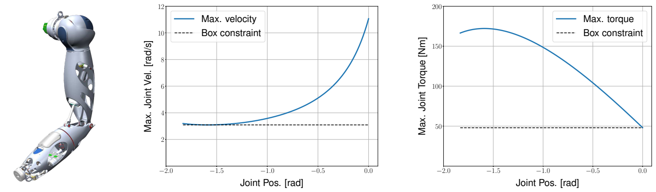


Fig. 2. Upper body of RH5 Manus robot



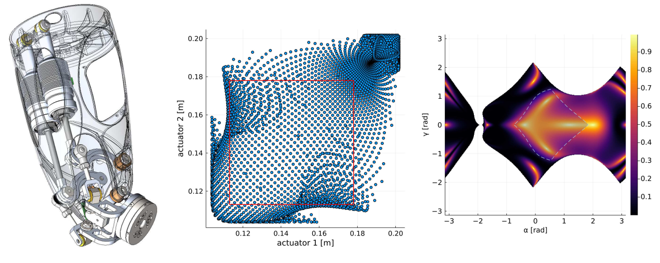
(a) 2SP̲U+1U (b) Actuation space (c) Independent joint space

Fig. 3. Box constraints for RH5 Manus torso [30].



(a) 1-RRP̲R (b) Position vs. max. vel. (c) Position vs. max. torque

Fig. 4. Box constraints for velocities/torques for RH5 Manus elbow [29].



(a) 2SURRP̲R+1U (b) Actuation space (c) Independent joint space

Fig. 5. Box constraints for RH5 Manus wrist [27].

TABLE II
ROM OF THE INDEPENDENT JOINTS IN RH5 MANUS

Joint	ROM (°)	Max. Torque (N m)	Max. Velocity (°/s)
Elbow	$[-105°, 0°]$	18.98–67.96	177–633
Wrist Pitch	$[-42.5°, 100°]$	29–56	364–696
Wrist Yaw	$[-32°, 34°]$	38–50	386–499
Torso Pitch	$[-20°, 30°]$	107.45–141.41	184–238
Torso Roll	$[-25.5°, 25.5°]$	91.42–110.67	208–400

C. Tree Abstraction of RH5 Manus

As multiple closed-loop mechanisms in a robot can become complex to resolve for errors in a computationally efficient manner, a tree-abstraction of the whole system is generally considered. A tree abstraction of the RH5 Manus robot in Fig. 2, is a tree-type system considering only independent coordinated or green edges of the topological graph. The tree abstraction of RH5 Manus robot consist of 20 degrees of freedom $(m = 20)$ and Table II provides the joint limits for the closed-loop mechanisms. Note that a conservative choice would be to consider lower maximum values for torques which may underestimate the capabilities of the robot. An ambitious choice would be to consider the upper maximum limits for torques which would overestimate the capabilities of the robot.

D. Optimal control formulation

The optimization problem was designed with the same costs and time horizon for the full hybrid model and the tree-type abstraction model, with 3 different choice of torque and velocity limits, namely: lower, middle and upper limits. The middle limits for torque and velocity of the joint are computed as the average of the lower and upper limits. For the full robot model, 61 DOF model with 30 loop-closure constraints were considered in the optimization process. These constraints are defined in the SDF model and automatically mapped by the Pinocchio software. For the open tree-type model, 20 DOFs with no kinematic constraints were involved in the optimization. The optimal trajectories are computed numerically by solving the Optimal Control Problem (OCP) using Crocoddyl. As a result, we can obtain the decision variables: state $\boldsymbol{x} = (\mathbf{q}, \dot{\mathbf{q}})$ and control command $(\boldsymbol{\tau})$. For the motions of the constrained model, the forces ϕ_k are calculated in accordance with the constrained dynamics formulated in (3). The optimal trajectories for the task movements are formulated with a running cost model and a terminal cost model. The running cost models are defined by the following cost functions:

$$l = \sum_{c=1}^{C} \alpha_c \Phi_c(\mathbf{q}, \dot{\mathbf{q}}, \tau), \tag{5}$$

$\alpha_c \in \mathbb{R}$ is the applied weight to the cost function Φ_c. All the generated motions involves the same type of cost functions.

- *Wrist Target tracking*: The wrist position r_w (right and left) track the final wrist target placement for each desired end configuration.
  $\Phi_1 = \| r_w(t) - r_w^{ref}(t_N) \|_2^2 \, \alpha_1$
- *Control regularization*: Minimization of the joint control for dynamically feasible motions.
  $\Phi_2 = \| \tau(t) \|_2^2 \, \alpha_2$
- *Posture regularization*: This cost manage the redundancy of the multi-body system.
  $\Phi_3 = \| \mathbf{q}(t) - \mathbf{q}^{ref}(t_N) \|_2^2 \, \alpha_3$

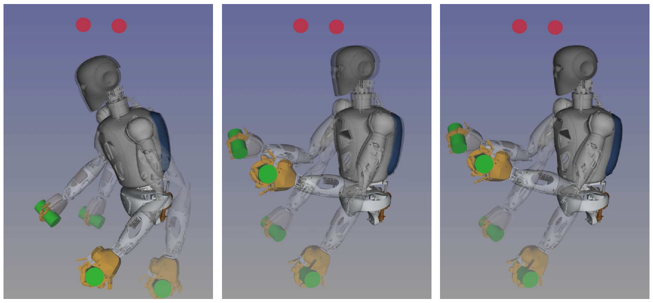
Here, t_N refers to the final time of the motion. The terminal cost model includes only the wrist target tracking and the posture regularization cost. In addition to these cost functions, in the full model formulation of the OCP, the proximal parameter has to be determined as well. In this 15 kg lifting simulation, the proximal parameter $prox = 10^{-5}$. All the hyper-parameters in the OCP formulations are determined empirically.

IV. Results

This section presents the results of trajectory optimization on a fixed-base RH5 Manus robot for the weight-lifting task[1]. The chosen weight for trajectory optimization is 15 kg (7.5 kg in each hand). These trajectories are first presented in simulation and also tested on the real robot. Lastly, the computational timings are reported for trajectory optimization in both the serial abstraction model and the full hybrid model of the robot. The simulation and experimental results can also be seen in the accompanying video.

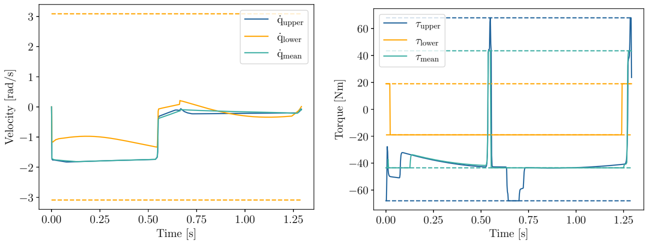
A. Tree abstraction model vs full hybrid model

For full hybrid model, the main objective is to feasible trajectory considering the internal loop-closure constraints in the robot. For the tree abstraction model, the closed loop kinematic constraints are neglected. The trajectory optimization formulation is done to lift 15 kg weight (7.5 kg in each arm). The formulation of OCPs based on the two models is set to reach the same end-effector targets while lifting the same weight. The same cost models and time horizons were applied for both models as defined in the previous section.



(a) Lower (b) Middle (c) Upper

Fig. 6. Screenshots of the simulations of RH5 Manus robot lifting 15 kg with the serial abstraction model for lower, middle and upper limits. The blurred images correspond to the initial state of the robot and red bubbles denote the end-effector targets.



(a) Left elbow joint velocities (b) Left elbow joint torques

Fig. 7. Independent joint velocity and torque obtained from the simulations of lifting 15kg using the tree-type model (dotted lines denote the limits).

1) Tree abstraction model: The trajectory optimization results are shown in Fig. 6 for the considered three cases. It can be seen that the tree-type model of the robot is not able to reach the defined target (red bubbles in Fig.6) in all the three cases. It performs badly with lower and middle limits of the independent joints of the closed-loop mechanism. The trajectory optimization with the upper maximum limits on the independent joints performs better than the other two cases and is closer to the desired target.

In Fig. 7(b), same can be observed from the torques plot for left elbow joint from trajectory optimization. The dotted lines in the plot refers to the joint torque limits for the three cases. For lower and middle maximum limits, the left elbow joint torques are getting saturated during the trajectory. However, with upper maximum limits, elbow torques are less saturated. Considering conservative limits leads to task failure in the trajectory optimization. On the other hand, ambitious choice of the limits may find results in some cases but are not transferable on the robot.

To support the argument on ambitious choice of limits in independent joint configuration space, the results for upper maximum limits are observed in actuation space of the closed-loop mechanism. The torque results from trajectory optimization are mapped in actuation space using the inverse statics mapping of the mechanism. The linear actuator

[1] `https://github.com/dfki-ric-underactuated-lab/case_study_traj_opt_hybrid_robots`, `https://youtu.be/qcJiLLTbDmk`

forces required to perform the obtained trajectories from optimization are plotted in Fig. 9(a). It can be observed that the mapped actuation forces are outside the real limits for both linear actuators in left and right elbow closed-loop mechanisms. Therefore, trajectory optimization while considering the serial abstraction model of the robot failed to perform for the required task.

2) Full hybrid model: In the full hybrid model of the robot, full actuation capabilities of the robot are exploited by planning directly in the actuation space of the robot. OCP formulation remains same for trajectory optimization as discussed before. The results are shown in Fig. 8 where the robot can be seen to reach the desired target. The torque plots in the actuation space can be seen in Fig. 9(b). For both linear actuators in the elbow closed-loop mechanisms, the forces are well under the actuator limits. The trajectory optimization is able to find feasible trajectories to lift 15 kg weight in full hybrid model as opposed to serial model. Hence, considering full hybrid model provides an edge over serial abstraction model in trajectory optimization by exploiting admissible configuration space in a proper manner.

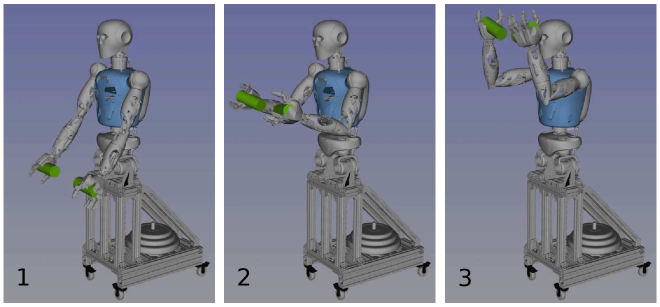


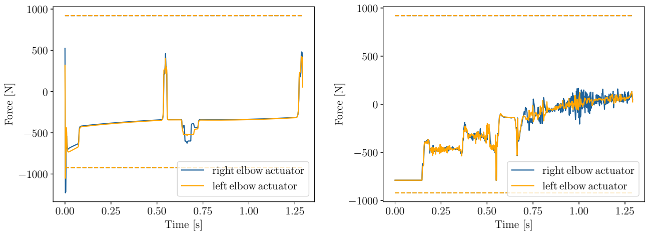
Fig. 8. Screenshots of the obtained simulations of the robot lifting 15 kg weight with full hybrid model.

B. Experimental Results

In addition to the presented simulations, the motion was also performed on the real robot. To achieve the obtained full hybrid models motion on the system, the independent joint position trajectories are mapped to the actuator space using HyRoDyn [32]. The snapshots of the robot executing the motion can be seen in Fig. 1. The velocity and effort plots for the left elbow actuator are shown in Fig. 10. The plots highlight the fact that the reference trajectories obtained from trajectory optimization, while considering a full hybrid robot can also be performed on the real robot with velocity and effort respecting the limits.

C. Computational Timings

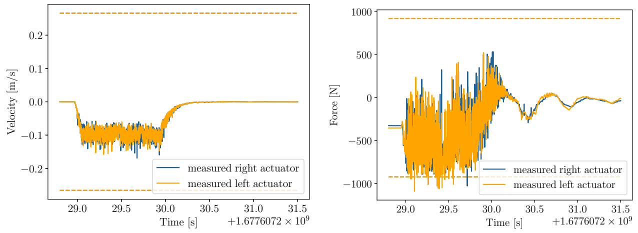
The average CPU time was measured for solving the OCP for 1 iteration 1000 times on a standard laptop with an Intel(R) Core(TM) i7-10750H @ 2.60GHz processor. These were 31.07 ms and 412.27 ms respectively for tree-type model and full hybrid model which show a large difference when the closed loops constraints are included in the resolution of an OCP. For the OCP resolution formulated in this work, the computational time using the tree-type model was 3.86 seconds, while for the full hybrid model, the trajectory optimization required about 1.5 hours.



(a) Forces mapped from tree abstraction model (upper limits)

(b) Forces computed from full hybrid model

Fig. 9. Actuation forces obtained in simulations for the elbow mechanism.



(a) Velocity

(b) Force

Fig. 10. Elbow actuators velocity and effort for lifting 15 kg weight on the real system.

V. Conclusion and Outlook

In this work, we generated a dynamic weight-lifting motion using the design specifications of the RH5 Manus robot model. The RH5 Manus, with its series-parallel hybrid design, provides high stiffness and large payload lifting capabilities. However, in order to exploit these capabilities, it is important to take into account the full robot model, as demonstrated in the paper. For this purpose, we included the loop-closures constraints in the trajectory optimization process. This allowed us to overcome the limitations of the tree-like abstraction model currently used in the state of the art to model and control systems involving loop closure mechanisms. Through this process, we were able to achieve 15 Kg weight lifting motions in simulation and on the real system. Thus, not only the payload capabilities are improved using the constrained full model, but also the workspace of the robot is better exploited. This feature was highlighted in the weight lifting motion of 15 Kg, where neither of the three tree-like abstraction models could successfully achieve an optimal trajectory to reach the target configuration while lifting the loads whereas the full model succeeded in this task. This work allows us to demonstrate a well-known theoretical property, that exploiting the closed loop kinematics constraints in the trajectory optimization process leads to better trajectories for the series-parallel hybrid robots. The closed loops were modeled using implicit constraints at the acceleration level which are susceptible to numerical inaccuracies. In order to ensure that these do not affect the physical robot in a negative way, the resulting trajectories were verified with an explicit solution of the constraints using HyRoDyn software [32] before sending them to the robot. Additionally, it was noted that tuning this kind of constrained OCP requires selecting a suitable proximal parameter value. This parameter is variable from task to task for the same robot model. The adjustment of all hyper-parameters in the constrained OCP is more sensitive than a classical OCP and can be time consuming. The computational time of solving a constrained OCP also increases significantly, making online stabilization, including loop-closure impossible at the moment. A bi-level optimization could be a solution to achieve online trajectory optimization while respecting all the capabilities of the robot. This approach should involve different time horizons for each optimization level, separating the resolution of the loop-closures from the minimization of the cost model. To avoid such numerical and computational efficiency issues, explicit formulation of loop-closures could be implemented in the OCP. Furthermore, we would also like to extend the HyRoDyn software to include trajectory optimization in order to resolve the loop-closures constraints in an explicit form.

References

[1] S. Kumar, H. Whrle, M. Trampler, M. Simnofske, H. Peters, M. Mallwitz, E. A. Kirchner, and F. Kirchner, "Modular design and decentralized control of the recupera exoskeleton for stroke rehabilitation," *Applied Sciences*, vol. 9, no. 4, 2019. [Online]. Available: https://www.mdpi.com/2076-3417/9/4/626

[2] I. Tijjani, S. Kumar, and M. Boukheddimi, "A survey on design and control of lower extremity exoskeletons for bipedal walking," *Applied Sciences*, vol. 12, no. 5, p. 2395, 2022.

[3] D. Kuehn, M. Schilling, T. Stark, M. Zenzes, and F. Kirchner, "System design and testing of the hominid robot charlie," *Journal of Field Robotics*, vol. 34, no. 4, pp. 666–703, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21662

[4] N. A. Radford, P. Strawser, K. Hambuchen, J. Mehling, and et. al., "Valkyrie: Nasa's first bipedal humanoid robot," *Journal of Field Robotics*, vol. 32, no. 3, pp. 397–419, 2015.

[5] J. Esser, S. Kumar, H. Peters, V. Bargsten, J. d. G. Fernandez, C. Mastalli, O. Stasse, and F. Kirchner, "Design, analysis and control of the series-parallel hybrid rh5 humanoid robot," in *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, 2021, pp. 400–407.

[6] S. Kumar, H. Wöhrle, J. de Gea Fernández, A. Müller, and F. Kirchner, "A survey on modularity and distributivity in series-parallel hybrid robots," *Mechatronics*, vol. 68, p. 102367, 2020.

[7] Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4906–4913.

[8] K. Mombaur, "Using optimization to create self-stable human-like running," *Robotica*, vol. 27, no. 3, p. 321330, 2009.

[9] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Trans. Graph.*, vol. 31, no. 4, jul 2012.

[10] W. Xi and C. D. Remy, "Optimal gaits and motions for legged robots," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3259–3265.

[11] H. Guo, Y. Liu, G. Liu, and H. Li, "Cascade control of a hydraulically driven 6-dof parallel robot manipulator based on a sliding mode," *Control Engineering Practice*, vol. 16, no. 9, pp. 1055–1068, 2008.

[12] D. Pri, N. Nedi, and V. Stojanovi, "A nature inspired optimal control of pneumatic-driven parallel robot platform," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 231, no. 1, pp. 59–71, 2017. [Online]. Available: https://doi.org/10.1177/0954406216662367

[13] N. Nedic, V. Stojanovic, and V. Djordjevic, "Optimal control of hydraulically driven parallel robot platform based on firefly algorithm," *Nonlinear Dynamics*, vol. 82, no. 3, pp. 1457–1473, 2015.

[14] M. Posa, S. Kuindersma, and R. Tedrake, "Optimization and stabilization of trajectories for constrained dynamical systems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1366–1373.

[15] C. Mastalli, R. Budhiraja, W. Merkt, G. Saurel, B. Hammoud, M. Naveau, J. Carpentier, S. Vijayakumar, and N. Mansard, "Crocoddyl: An Efficient and Versatile Framework for Multi-Contact Optimal Control," in *ICRA*, 2020.

[16] M. Boukheddimi, S. Kumar, H. Peters, D. Mronga, R. Budhiraja, and F. Kirchner, "Introducing RH5 Manus: A Powerful Humanoid Upper Body Design for Dynamic Movements," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8540–8546.

[17] M. Boukheddimi, D. Harnack, S. Kumar, R. Kumar, S. Vyas, O. Arriaga, and F. Kirchner, "Robot dance generation with music based trajectory optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3069–3076.

[18] S. Kumar, J. Martensen, A. Mueller, and F. Kirchner, "Model simplification for dynamic control of series-parallel hybrid robots - a representative study on the effects of neglected dynamics shivesh," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 5701–5708.

[19] J. Carpentier, R. Budhiraja, and N. Mansard, "Proximal and Sparse Resolution of Constrained Dynamic Equations," in *Robotics: Science and Systems 2021*, Austin / Virtual, United States, July 2021. [Online]. Available: https://hal.inria.fr/hal-03271811

[20] R. Featherstone, *Rigid body dynamics algorithms*. Springer, 2014.

[21] 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.

[22] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *SII*. IEEE, 2019.

[23] N. Mansard, "Feasibility-prone differential dynamic programming is ddp a multiple shooting algorithm?" Online, 2020, available: https://github.com/loco-3d/crocoddyl/tree/master/doc/reports/fddp.

[24] R. Budhiraja, J. Carpentier, C. Mastalli, and N. Mansard, "Differential dynamic programming for multi-phase rigid contact dynamics," in *Humanoids*. IEEE, 2018.

[25] C. Mastalli, W. Merkt, J. Marti-Saumell, H. Ferrolho, , J. Sola, N. Mansard, and S. Vijayakumar, "A direct-indirect hybridization approach to control-limited ddp," *arXiv:2010.00411*, 2021.

[26] S. Kumar, A. Nayak, H. Peters, C. Schulz, A. Müller, and F. Kirchner, "Kinematic analysis of a novel parallel 2sprr+1u ankle mechanism in humanoid robot," in *Advances in Robot Kinematics 2018*, J. Lenarcic and V. Parenti-Castelli, Eds. Cham: Springer International Publishing, 2019, pp. 431–439.

[27] C. Stoeffler, A. del Rio Fernandez, H. Peters, M. Schilling, and S. Kumar, "Kinematic Analysis of a Novel Humanoid Wrist Parallel Mechanism," in *Advances in Robot Kinematics 2022*, O. Altuzarra and A. Kecskeméthy, Eds. Cham: Springer International Publishing, 2022, pp. 348–355.

[28] Y. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *ICRA*. IEEE, 2014, pp. 1168–1175.

[29] D. Mronga, S. Kumar, and F. Kirchner, "Whole-body control of series-parallel hybrid robots," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 228–234.

[30] S. Kumar, "Modular and analytical methods for solving kinematics and dynamics of series-parallel hybrid robots," Ph.D. dissertation, Universität Bremen, 2019.

[31] R. Kumar, S. Kumar, A. Müller, and F. Kirchner, "Modular and hybrid numerical-analytical approach - a case study on improving computational efficiency for series-parallel hybrid robots," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 3476–3483.

[32] S. Kumar, K. A. v. Szadkowski, A. Mueller, and F. Kirchner, "An analytical and modular software workbench for solving kinematics and dynamics of series-parallel hybrid robots," *Journal of Mechanisms and Robotics*, vol. 12, no. 2, 2020.