APPENDIX

A. Joint Limits of Loco-Manipulator

The range of motion for loco-manipulator's joints is illustrated in Fig. 12, with angles of 280°, 210°, and 360° for the first through the final joint, respectively.

The motion limitation of the first joint arises from a collision between the first servo and the support structure that is part of the outer calf connector (Fig. 2). This support structure is designed primarily to connect to the opposite horn of the first servo, playing a vital role in stabilizing the manipulator. The design of the support structure aims to minimize its impact on the essential range of motion of the first servo. This critical range is necessary for extending the gripper during transitions from locomotion to manipulation and for positioning the gripper forwards and towards the ground, representing the primary workspace. The limitations in the motion of the second joint are due to self-collision. For the third joint, it does not face stringent motion restrictions, except for those imposed by the cable's length.

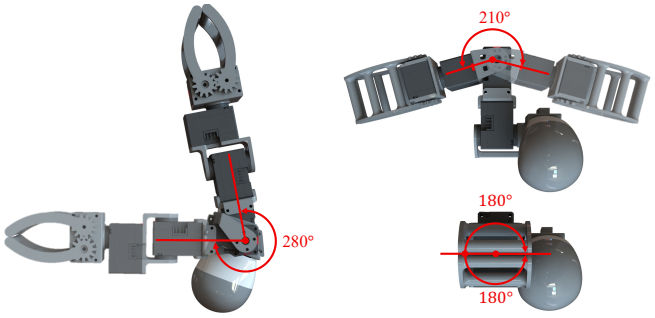


Figure 12: The range of motion for loco-manipulator's joints. From the first joint to the final one, the corresponding angular movements are 280°, 210°, and 360° respectively.

B. Whole-Body Impulse Control

Notation from IV: robot state $\boldsymbol{q} = (\boldsymbol{x}^{\text{torso}}, \boldsymbol{q}_j) \in \mathbb{R}^{24}$, where $\boldsymbol{x}^{\text{torso}} \in \mathbb{R}^6$ denotes the position and orientation of the floating torso, and $\boldsymbol{q}_j \in \mathbb{R}^{18}$ denotes the joint angles of LocoMan. The bar notation $\bar{(\cdot)}$ denotes all desired states.

*1) **Hierarchical Objective Tracking**:* As depicted in Table III, there are n tracking objectives in different operation modes, comprising n tracking tasks to be executed. The null space projection technique is utilized to track each objective in hierarchical order (from 1-st to n-th).

In the i-th ($i = 1, ..., n$) task, the tracking items consist of the desired position $\overline{\boldsymbol{x}_i}$, velocity $\overline{\dot{\boldsymbol{x}}_i}$, and acceleration $\overline{\ddot{\boldsymbol{x}}_i}$ of the i-th tracking objective, which are tracked by computing the desired position $\overline{\boldsymbol{q}}_i$, velocity $\overline{\dot{\boldsymbol{q}}}_i$, and acceleration $\overline{\ddot{\boldsymbol{q}}}_i$ for both the floating torso and the joints. The desired states ($\overline{\boldsymbol{q}}_n$, $\overline{\dot{\boldsymbol{q}}}_n$, and $\overline{\ddot{\boldsymbol{q}}}_n$) for the last tracking objective would be the final desired states ($\overline{\boldsymbol{q}}$, $\overline{\dot{\boldsymbol{q}}}$, and $\overline{\ddot{\boldsymbol{q}}}$) of LocoMan.

$$
\begin{aligned}
\overline{\boldsymbol{q}_i} &= \overline{\boldsymbol{q}_{i-1}} + \boldsymbol{J}_{i|pre}{}^{\dagger} \left(\overline{\boldsymbol{x}_i} - \boldsymbol{x}_i - \boldsymbol{J}_i \left(\overline{\boldsymbol{q}_{i-1}} - \boldsymbol{q}\right)\right) \\
\overline{\dot{\boldsymbol{q}}_i} &= \overline{\dot{\boldsymbol{q}}_{i-1}} + \boldsymbol{J}_{i|pre}{}^{\dagger} \left(\overline{\dot{\boldsymbol{x}}_i} - \boldsymbol{J}_i \overline{\dot{\boldsymbol{q}}_{i-1}}\right) \\
\overline{\ddot{\boldsymbol{q}}_i} &= \boldsymbol{J}_{i|pre}^{\text{dyn}\ \ddagger} \left(\ddot{\boldsymbol{x}}_i^{\text{cmd}} - \dot{\boldsymbol{J}}_i \dot{\boldsymbol{q}} - \boldsymbol{J}_i \overline{\ddot{\boldsymbol{q}}_{i-1}}\right)
\end{aligned} \tag{1}
$$

The acceleration command $\ddot{\boldsymbol{x}}_i^{\text{cmd}}$ is computed by:

$$
\ddot{\boldsymbol{x}}_i^{\text{cmd}} = \overline{\ddot{\boldsymbol{x}}_i} + \boldsymbol{K}_p^{acc} \left(\overline{\boldsymbol{x}_i} - \boldsymbol{x}_i\right) + \boldsymbol{K}_d^{acc} \left(\overline{\dot{\boldsymbol{x}}_i} - \dot{\boldsymbol{x}}_i\right) \tag{2}
$$

where $\boldsymbol{K}_p^{acc}$ and $\boldsymbol{K}_d^{acc}$ denote position and velocity feedback gains.

To compute the desired states of LocoMan from the tracking objective, we need to use the pseudo-inverse of the Jacobian. In Equation (1), we track the desired position and velocity kinematically, and the desired acceleration dynamically. Therefore, two types of pseudo-inverses are utilized. One is the SVD-based pseudo-inverse denoted by $\{\cdot\}^{\dagger}$, whcih is defined as:

$$
\begin{aligned}
\boldsymbol{J} &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \\
\boldsymbol{J}^{\dagger} &= \boldsymbol{V}\boldsymbol{\Sigma}^{\dagger}\boldsymbol{U}^T
\end{aligned} \tag{3}
$$

where $\boldsymbol{J}$ is the Jacobian and $\boldsymbol{U}$ and $\boldsymbol{V}$ are the orthogonal matrices. Besides, $\boldsymbol{\Sigma}$ is the diagonal matrix, and $\boldsymbol{\Sigma}^{\dagger}$ is formed by taking the reciprocal of each non-zero element on the diagonal of $\boldsymbol{\Sigma}$, leaving the zeros in place, and then transposing the matrix.

The other is the dynamically consistent pseudo-inverse denoted by $\{\cdot\}^{\ddagger}$ with the definition of:

$$
\boldsymbol{J}^{\ddagger} = \boldsymbol{A}^{-1}\boldsymbol{J}^{\top} \left(\boldsymbol{J}\boldsymbol{A}^{-1}\boldsymbol{J}^{\top}\right)^{-1} \tag{4}
$$

where $\boldsymbol{A}$ is the mass matrix.

Furthermore, $\boldsymbol{J}_i$ is the Jacobian of the i-th tracking objective, while $\boldsymbol{J}_{i|pre}$ is the projection of it into the null space of all previous tasks. And the symbol $\{\cdot\}^{\text{dyn}}$ denotes the dynamically consistent constraints. $\boldsymbol{J}_{i|pre}$ and $\boldsymbol{J}_{i|pre}^{\text{dyn}}$ are computed with the following steps:

$$
\begin{aligned}
\boldsymbol{N}_{i-1} &= \boldsymbol{N}_0 \boldsymbol{N}_{1|0} \cdots \boldsymbol{N}_{i-1|i-2} \\
\boldsymbol{J}_{i|pre} &= \boldsymbol{J}_i \boldsymbol{N}_{i-1} \\
\boldsymbol{J}_{i|i-1} &= \boldsymbol{J}_i \left(\boldsymbol{I} - \boldsymbol{J}_{i-1}{}^{\dagger} \boldsymbol{J}_{i-1}\right) \\
\boldsymbol{N}_{i|i-1} &= \boldsymbol{I} - \boldsymbol{J}_{i|i-1}{}^{\dagger} \boldsymbol{J}_{i|i-1} \\
\boldsymbol{N}_{i-1}^{\text{dyn}} &= \boldsymbol{N}_0^{\text{dyn}} \boldsymbol{N}_{1|0}^{\text{dyn}} \cdots \boldsymbol{N}_{i-1|i-2}^{\text{dyn}} \\
\boldsymbol{J}_{i|pre}^{\text{dyn}} &= \boldsymbol{J}_i \boldsymbol{N}_{i-1}^{\text{dyn}} \\
\boldsymbol{J}_{i|i-1}^{\text{dyn}} &= \boldsymbol{J}_i \left(\boldsymbol{I} - \boldsymbol{J}_{i-1}{}^{\ddagger} \boldsymbol{J}_{i-1}\right) \\
\boldsymbol{N}_{i|i-1}^{\text{dyn}} &= \boldsymbol{I} - \boldsymbol{J}_{i|i-1}^{\text{dyn}\ \ddagger} \boldsymbol{J}_{i|i-1}^{\text{dyn}}
\end{aligned} \tag{5}
$$

where $\boldsymbol{N}_{i-1}$, $\boldsymbol{J}_{i|i-1}$, and $\boldsymbol{N}_{i|i-1}$ represent the null space of the previous $(i-1)$ tasks, the projection of the Jacobian $\boldsymbol{J}_i$ into the null space of the $(i-1)$-th task, and the independent null space introduced by the i-th task.

Here, i is within the range of $1, ..., n$. And the initial items are defined as:

$$
\begin{aligned}
\boldsymbol{N}_0 &= \boldsymbol{I} - \boldsymbol{J}_c{}^{\dagger} \boldsymbol{J}_c \\
\boldsymbol{N}_0^{\text{dyn}} &= \boldsymbol{I} - \boldsymbol{J}_c{}^{\ddagger} \boldsymbol{J}_c \\
\boldsymbol{J}_0 &= \boldsymbol{J}_c \\
\overline{\boldsymbol{q}_0} &= \boldsymbol{q} \\
\overline{\dot{\boldsymbol{q}}_0} &= \boldsymbol{0} \\
\overline{\ddot{\boldsymbol{q}}_0} &= \boldsymbol{J}_c{}^{\ddagger} \left(-\boldsymbol{J}_c \dot{\boldsymbol{q}}\right)
\end{aligned} \tag{6}
$$

where $\boldsymbol{J}_c$ is the Jacobian of the feet in contact.

*2) **Quadratic Programming**:* With the desired acceleration $\overline{\ddot{\boldsymbol{q}}}$ computed in the previous step, we solve a Quadratic Program (QP) to optimize for ground reaction forces. The QP problem is formulated as:

$$\min_{\mathbf{f}_r, \boldsymbol{\delta}_t} \mathbf{f}_r^\top \boldsymbol{Q}_1 \mathbf{f}_r + \boldsymbol{\delta}_t^\top \boldsymbol{Q}_2 \boldsymbol{\delta}_t \tag{7}$$

$$\text{s.t. } \boldsymbol{S}_f(\boldsymbol{A}\ddot{\mathbf{q}}^{\text{cmd}} + \mathbf{b} + \mathbf{g}) = \boldsymbol{S}_f \boldsymbol{J}_c^\top \mathbf{f}_r \quad \text{(floating torso dyn.)}$$

$$\ddot{\mathbf{q}}^{\text{cmd}} = \overline{\ddot{\boldsymbol{q}}} + \begin{bmatrix} \boldsymbol{\delta}_t \\ \mathbf{0}_j \end{bmatrix} \quad \text{(floating torso acceleration)}$$

$$\boldsymbol{W}\mathbf{f}_r \geq \mathbf{0} \quad \text{(contact force constraints)}$$

where the optimization objective is to minimize the reaction force $\mathbf{f}_r$ and the relaxation variable $\boldsymbol{\delta}_t$ for the floating torso acceleration with the weights $\boldsymbol{Q}_1$ and $\boldsymbol{Q}_2$. $\boldsymbol{S}_f$, $\boldsymbol{A}$, $\mathbf{b}$, $\mathbf{g}$, and $\boldsymbol{W}$ are the floating torso selection matrix, the mass matrix, the Coriolis force, the gravitation force, and the force constraint matrix for foot contact, respectively.

With the optimized reaction force $\mathbf{f}_r$ and relaxation variable $\boldsymbol{\delta}_t$ for the floating torso acceleration, we compute the joint torque command with the multi-body dynamics formulation:

$$\begin{bmatrix} \overline{\boldsymbol{\tau}_t} \\ \overline{\boldsymbol{\tau}_j} \end{bmatrix} = A\ddot{\mathbf{q}}^{\text{cmd}} + \mathbf{b} + \mathbf{g} - \boldsymbol{J}_c^\top \mathbf{f}_r \tag{8}$$

where $\boldsymbol{\tau}_j$ is the torque command for LocoMan's joints.

*3) **Joint-Level PD Control**:* We compute the desired position $\overline{\boldsymbol{q}_j}$, velocity $\overline{\dot{\boldsymbol{q}}_j}$, and torque $\overline{\boldsymbol{\tau}_j}$ for LocoMan's joints in the previous steps, which runs in 400Hz. The final torque command $\boldsymbol{\tau}_j$ applied to each joint is computed by a joint-level PD controller that runs in 2000Hz:

$$\boldsymbol{\tau}_j = \overline{\boldsymbol{\tau}_j} + \boldsymbol{K}_p^{tau}\left(\overline{\boldsymbol{q}_j} - \boldsymbol{q}_i\right) + \boldsymbol{K}_d^{tau}\left(\overline{\dot{\boldsymbol{q}}_j} - \dot{\boldsymbol{q}}_i\right) \tag{9}$$

The high frequency joint-level PD feedback enables the robot to track desired poses smoothly and accurately.

*4) **PD Gains for Real Robot**:* In the experiments described in V, we tune the PD gains of the tracking objectives for the whole-body impulse controller (Table V) and the PD gains of the hip, thigh, and calf joints for the joint-level PD controller (Table VI). We use the SDK from Dynamixel® and set the position gain values to $[8, 6, 6]^T \times 128$ and the velocity gain values to $[65, 40, 40]^T \times 16$ for the three joints of loco-manipulator in all operation modes, where 128 and 16 are the default scales in the SDK.

Table V: PD gains of the tracking objectives for computing the acceleration errors in whole-body impulse control.

Tracking Objective	$\boldsymbol{K}_p^{acc}$	$\boldsymbol{K}_d^{acc}$
Torso Position (Locomotion)	$[100, 100, 100]^T$	$[10, 10, 10]^T$
Torso Orientation (Locomotion)	$[100, 100, 100]^T$	$[10, 10, 10]^T$
Torso Position (Manipulation)	$[100, 100, 100]^T$	$[1, 1, 1]^T$
Torso Orientation (Manipulation)	$[100, 100, 100]^T$	$[1, 1, 1]^T$
Foot Position	$[100, 100, 100]^T$	$[10, 10, 10]^T$
Foot Orientation	$[100, 100, 100]^T$	$[10, 10, 10]^T$
Gripper Position	$[100, 100, 100]^T$	$[10, 10, 10]^T$
Gripper Orientation	$[100, 100, 100]^T$	$[10, 10, 10]^T$

Table VI: PD gains of the hip, thigh, and calf joints for computing the applied torque in joint-level PD control.

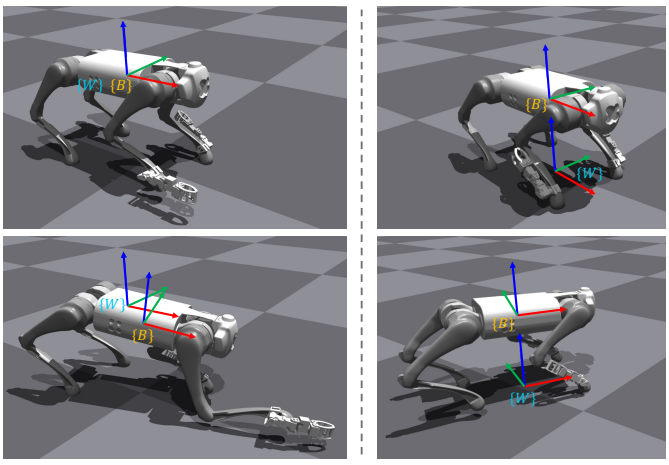
Leg Mode	$\boldsymbol{K}_p^{tau}$	$\boldsymbol{K}_d^{tau}$
Stance Legs in Locomotion	$[30, 30, 30]^T$	$[1, 1, 1]^T$
Swing Legs in Locomotion	$[30, 30, 30]^T$	$[1, 1, 1]^T$
Stance Legs in Loco-Manipulation	$[30, 30, 30]^T$	$[1, 1, 1]^T$
Swing Legs in Loco-Manipulation	$[20, 20, 20]^T$	$[0.8, 0.8, 0.8]^T$
Stance Legs in Single-Arm Manipulation	$[60, 60, 60]^T$	$[2, 2, 2]^T$
Swing Legs in Single-Arm Manipulation	$[60, 60, 60]^T$	$[2, 2, 2]^T$
Stance Legs in Bimanual Manipulation	$[100, 100, 100]^T$	$[2.5, 2.5, 2.5]^T$
Swing Legs in Bimanual Manipulation	$[30, 30, 30]^T$	$[1, 1, 1]^T$

C. State Estimation

As illustrated in Table III, there are two state estimators in our system, including the kinematics-based estimator used in the manipulation modes and the Kalman filter-based estimator used in the locomotion mode and the loco-manipulation mode.

As depicted in Fig. 13 (a), at the moment when LocoMan finishes the transition process from other operation modes to manipulation modes, the kinematics-based estimator sets the base frame of LocoMan as the world frame. Furthermore, we also record the positions of the contact foot as $\boldsymbol{P}_{init}^W$, which can be computed using forward kinematics and have the same values in the world frame or the base frame at the reset moment. During manipulation, the world frame remains fixed, and we compute the current positions of the contact foot in the base frame as $\boldsymbol{P}_{cur}^B$. Assuming that there is no slip occurring between the contact foot and the ground, we employ the Procrustes analysis method to solve the transform matrix from the world frame to the base frame $\boldsymbol{T}_W^B$.

As for locomotion, we implement the Kalman filter-based estimator which fuses the IMU information. As Fig. 13 (b) shows, the world frame is always updated to be the zero-yaw world frame.



(a) During Manipulation (b) During Locomotion

Figure 13: The definition of the world frame in different state estimators. (a) The kinematics-based estimator set the initial base frame of LocoMan as the world frame. (b) The Kalman filter-based estimator always has the zero-yaw world frame as the world frame.