

TUM Institute for Cognitive Systems (ICS)

OpenWalker

Module Description: Center of Mass (CoMM)

Emmanuel Dean, Florian Bergner, Rogelio Guadarrama-Olvera,
Simon Armleder, and Gordon Cheng

February 14, 2020

1 Module Description

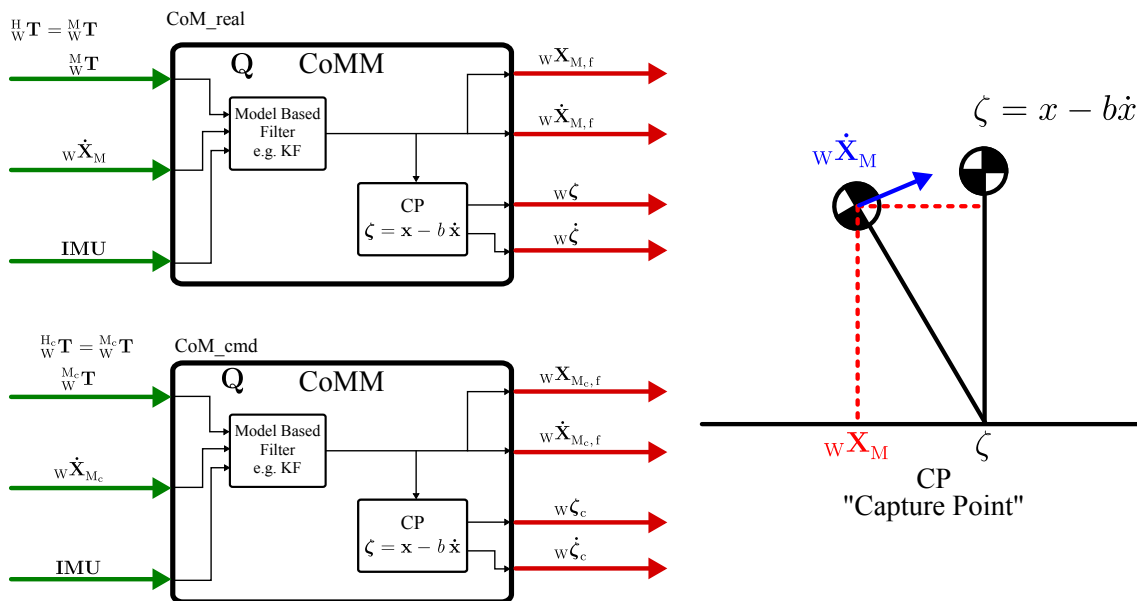


Figure 1.1: Center of Mass module: This module implements the center of mass estimation for the lower body of the robot, and computes the Capture point (CP) $w\zeta$.

The *Center of Mass* module (CoMM) estimates the state of the *Center of Mass* (CoM) of the lower body of the robot using a model-based estimator, e.g. Kalman Filter [2]. This estimator uses as an input the Cartesian position and velocity of the CoM obtained from the joint states

and the dynamic library (KDL/RBDL/KINDR), fused with the information from an IMU. The OpenWalker framework employs two CoMMs, one for computing the CoM of the real robot, and one for the commanded robot. The joint states of both robots can differ depending on the conditions of the environment.

The second task of this module is the real-time computation of the Divergent Component of Motion (DCM), also known as Capture Point (CP)[3]. The DCM is a major concept that has been applied to walking and running pattern generation. This CP must be computed for both the *real* and the *commanded* robot models. The CoM calculation of the *real* robot uses the Homogeneous transformation of the CoM *w.r.t* the *world* coordinate frame (wcf), ${}^M_W\mathbf{T}$, the Cartesian velocity of the CoM, ${}^W\dot{\mathbf{x}}_M$, and the IMU information¹. Correspondingly, the CoM of the *commanded* robot uses the commanded CoM position and velocity, ${}^{M_c}{}_W\mathbf{T}$ and ${}^W\dot{\mathbf{x}}_{M_c}$, respectively. The OpenWalker framework uses the *real* and *commanded* CoM state in the Balancer (BM) to produce adequate hip offsets to keep the balance of the robot.

Since the rigid multi body system (MBS) of the robot is the same for the *real* and the *commanded* robot the OpenWalker framework needs to realize only one CoMM, which is then implemented and connected to the *real* Cartesian state, and to the *commanded* Cartesian state.

2 Module Connections

2.1 Inputs

Symbol	Name	Type	Description
${}^M_W\mathbf{T} \in \mathbb{R}^{4 \times 4}$	Real Robot CoM Pose	HomogeneousTransformation	This matrix represents the pose of the real CoM with respect to the world coordinate frame (wcf).
${}^W\dot{\mathbf{x}}_M \in \mathbb{R}^6$	Real Robot CoM Velocities	CartesianVelocity	This vector contains the real CoM velocities of the robot.
$\text{IMU} \in \mathbb{R}^{10}$	IMU information	ImuSensor	This vector contains the IMU sensor information which includes Cartesian acceleration, angular position (in Quaternions), and angular velocity of the hip/torso.
${}^{M_c}{}_W\mathbf{T} \in \mathbb{R}^{4 \times 4}$	Commanded Robot CoM Pose	HomogeneousTransformation	This matrix represents the pose of the commanded CoM with respect to the world coordinate frame (wcf).
${}^W\dot{\mathbf{x}}_{M_c} \in \mathbb{R}^6$	Commanded Robot CoM Velocities	CartesianVelocity	This vector contains the commanded CoM velocities of the robot.
$\text{IMU} \in \mathbb{R}^{10}$	IMU information	ImuSensor	This vector contains the IMU sensor information which includes Cartesian acceleration, angular position (in Quaternions), and angular velocity of the hip/torso.

2.2 Outputs

CoM of the Real Robot

Symbol	Name	Type	Description
${}^W\mathbf{x}_{M,r} \in \mathbb{R}^6$	Real robot filtered CoM pose	CartesianPosition	This vector represents the filtered pose (position and orientation) of the real CoM.
${}^W\dot{\mathbf{x}}_{M,r} \in \mathbb{R}^6$	Real robot filtered CoM velocity	CartesianVelocity	This vector represents the filtered Cartesian velocity (linear and angular) of the real CoM.
${}^W\boldsymbol{\zeta} \in \mathbb{R}^3$	Real robot 3D Capture Point	DivergentComponentOfMotion	This vector represents the DCM position of the real robot.
${}^W\dot{\boldsymbol{\zeta}} \in \mathbb{R}^3$	Real robot 3D Capture Point velocity	DivergentComponentOfMotionP	This vector represents the DCM velocity of the real robot.

¹Usually the IMU sensor is located either in the hip or the torso of the robot.

CoM of the Commanded Robot

Symbol	Name	Type	Description
$w\hat{\mathbf{X}}_{M_c, f} \in \mathbb{R}^6$	Commanded Robot filtered CoM pose	CartesianPosition	This vector represents the filtered pose (position and orientation) of the commanded CoM.
$w\dot{\hat{\mathbf{X}}}_{M_c, f} \in \mathbb{R}^6$	Commanded Robot filtered CoM velocity	CartesianVelocity	This vector represents the filtered Cartesian velocity (linear and angular) of the commanded CoM.
$w\hat{\boldsymbol{\zeta}}_c \in \mathbb{R}^3$	Commanded robot 3D Capture Point	DivergentComponentOfMotion	This vector represents the DCM position of the commanded robot.
$w\dot{\hat{\boldsymbol{\zeta}}}_c \in \mathbb{R}^3$	Commanded robot 3D Capture Point velocity	DivergentComponentOfMotionP	This vector represents the DCM velocity of the commanded robot.

2.3 Inter-Connections

The inputs of the CoMM are connected to the outputs of the Forward Kinematic Model (FKM) which provides the pose and velocity of the CoM of the *real* and *commanded* robot. The CoMM input also is connected to the Real Robot Module (RRM) which provides a hardware interface for the IMU sensor. The outputs of CoMM (*real* and *commanded*) are directly connected to the Balancer module (BM).

2.4 Common Methods

The two main tasks of this module is the CoM state estimation and the computation of the DCM.

2.4.1 CoM Estimation

The general description of the state estimation is defined by [2]. This can be summarized as: Prediction of the state estimate:

$$\hat{\mathbf{X}}_{k|k-1} = \mathbf{F}_k \hat{\mathbf{X}}_{k|k-1} + \mathbf{B}_k \mathbf{u}_k \quad (2.1)$$

Prediction of the estimate covariance:

$$\mathbf{P}_{k|k-1} = \mathbf{F}_k \mathbf{P}_{k-1|k-1} \mathbf{F}_k^\top + \mathbf{Q}_k \quad (2.2)$$

Innovation of the residual:

$$\tilde{\mathbf{y}}_k = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k-1} \quad (2.3)$$

Innovation of the covariance:

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_k \quad (2.4)$$

Optimal Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \mathbf{S}_k^{-1} \quad (2.5)$$

update the state estimate

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{y}}_k \quad (2.6)$$

update of covariance:

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_{k|k-1} \quad (2.7)$$

with the updated residual as:

$$\tilde{\mathbf{y}}_{k|k} = \mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_{k|k} \quad (2.8)$$

2.4.2 DCM

The CP is a characteristic point of the linear inverted pendulum model. It was designed to address a question of push recovery: where should the robot step (instantaneously) to eliminate linear momentum ${}_m\dot{\mathbf{p}}_G$ and come (asymptotically) to a stop (See Fig. 1.1)? The general equation to compute the DCM is given by:

$${}_w\boldsymbol{\zeta} = {}_w\mathbf{x}_M - b{}_w\dot{\mathbf{x}}_M \quad (2.9)$$

where ${}_w\boldsymbol{\zeta} = {}_w[\zeta_x, \zeta_y, \zeta_z]^\top$ is the DCM, ${}_w\mathbf{x}_M = {}_w[x, y, z]^\top$ and ${}_w\dot{\mathbf{x}}_M = {}_w[\dot{x}, \dot{y}, \dot{z}]^\top$ are the CoM position and velocity and $b > 0$ is the time-constant of the DCM dynamics [3].

References

- [1] Martin L. Felis, RBDL: An efficient rigid-body dynamics library using recursive algorithms, *Autonomous Robots* 41 (2): 495–511, 2017.
- [2] R. E. Kalman, A New Approach to Linear Filtering and Prediction Problems, *Transaction of the ASME–Journal of Basic Engineering*, pp. 35-45, March 1960.
- [3] Engelsberger, Johannes, et al. Three-dimensional bipedal walking control based on divergent component of motion, *Ieee transactions on robotics*, pp. 355-368, 2015.