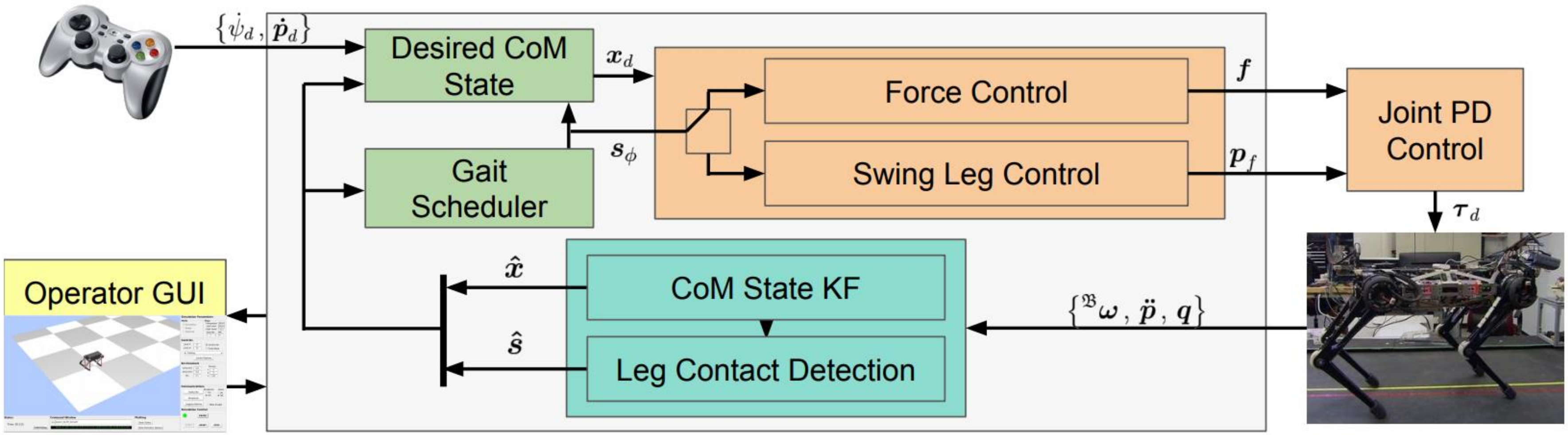
# The Control Architecture for MIT Cheetah 3

Block Diagram, Formulation and Implementation

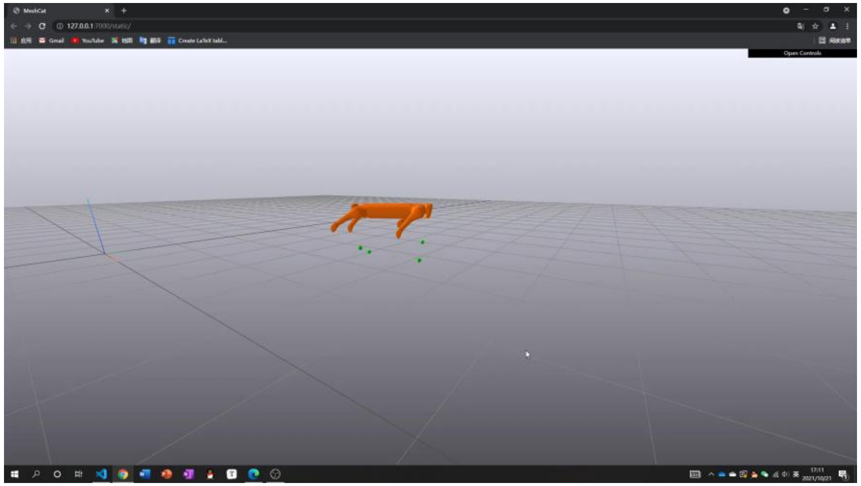
SUN Yinghan

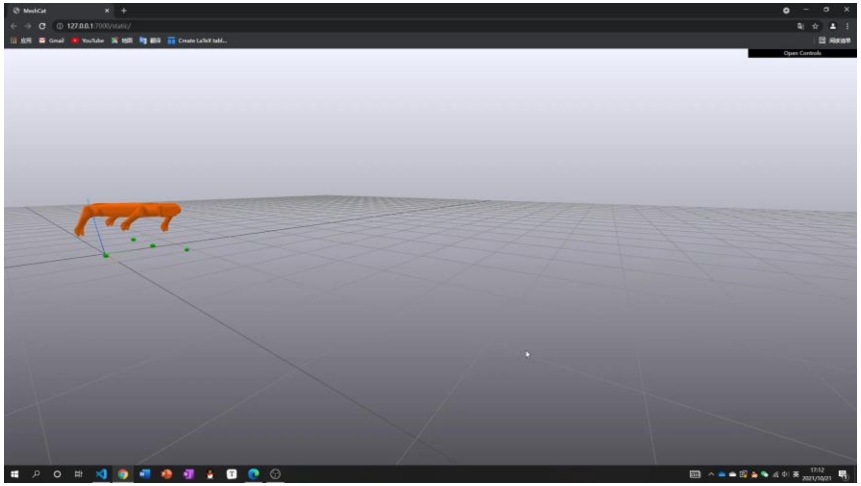
2021. 10. 22

# Overview



$\{\dot{\psi}_d, \dot{\boldsymbol{p}}_d\}$
Desired CoM State
$\boldsymbol{x}_d$
Gait Scheduler
$\boldsymbol{s}_\phi$
Force Control
Swing Leg Control
$\boldsymbol{f}$
$\boldsymbol{p}_f$
Joint PD Control
$\boldsymbol{\tau}_d$
Operator GUI
$\hat{\boldsymbol{x}}$
$\hat{\boldsymbol{s}}$
CoM State KF
Leg Contact Detection
$\{{}^{\mathfrak{B}}\boldsymbol{\omega}, \ddot{\boldsymbol{p}}, \boldsymbol{q}\}$



# Outline

**Part I:** Gait Scheduler

**Part II:** State Estimation

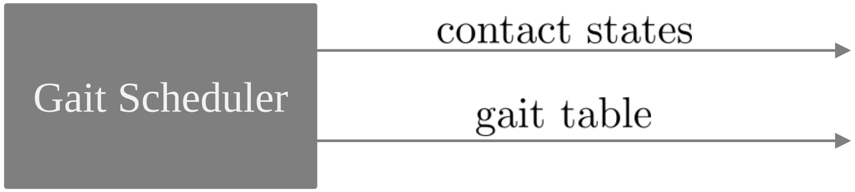
**Part III:** Body Control: Model-Predictive Controller

**Part IV:** Swing Leg Trajectory and Leg Control

**Part V:** Summary

# **Part I:** Gait Scheduler

# Gait Scheduler




**Data Structure:**

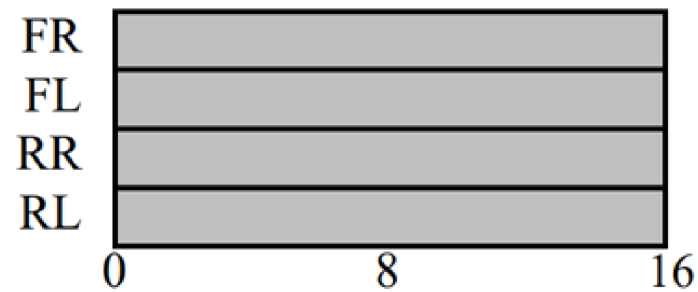
- swing time and stance time
- swing states and contact states
- gait table

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

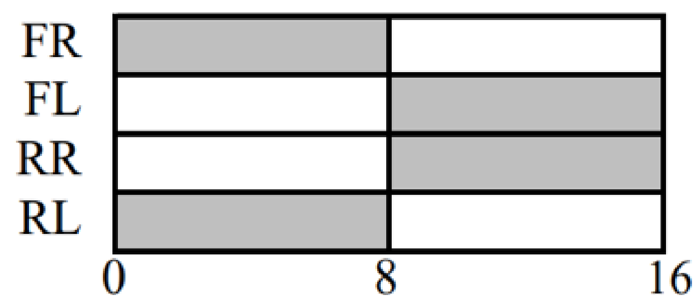
| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Define a gait:**

- number of segments
- offsets
- durations



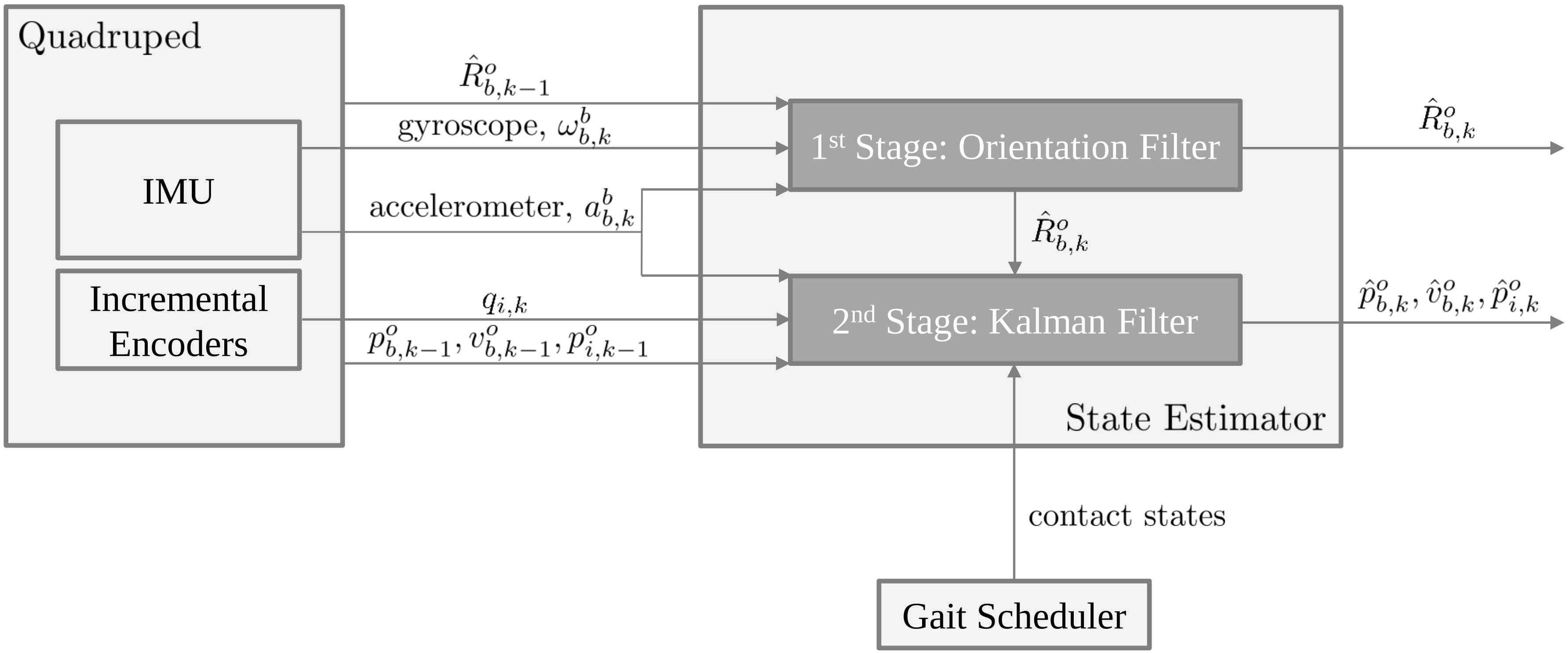

standing




trotting

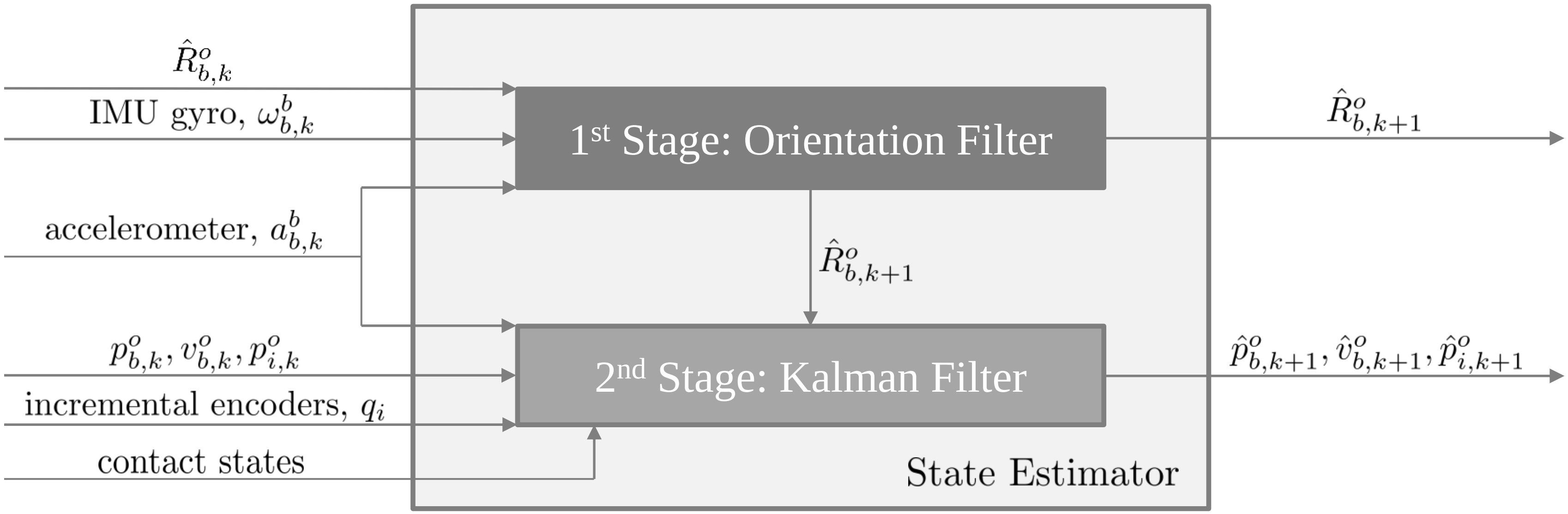
# **Part II:** State Estimation

# Framework



Quadruped
IMU
Incremental Encoders
$\hat{R}^o_{b,k-1}$
gyroscope, $\omega^b_{b,k}$
accelerometer, $a^b_{b,k}$
$q_{i,k}$
$p^o_{b,k-1}, v^o_{b,k-1}, p^o_{i,k-1}$
1st Stage: Orientation Filter
$\hat{R}^o_{b,k}$
2nd Stage: Kalman Filter
State Estimator
$\hat{R}^o_{b,k}$
$\hat{p}^o_{b,k}, \hat{v}^o_{b,k}, \hat{p}^o_{i,k}$
contact states
Gait Scheduler

# $1^{st}$-Stage: Orientation Filter



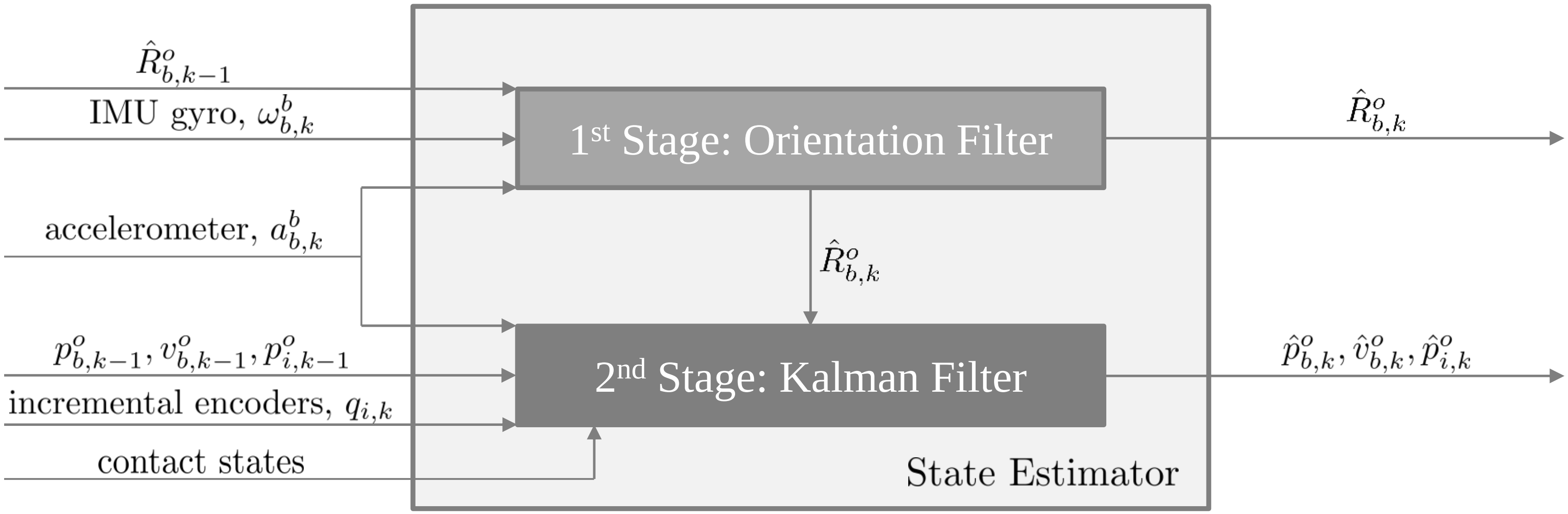

The filter updates the estimation of orientation according to

$$\hat{R}^o_{b,k+1} = \hat{R}^o_{b,k} \left[\omega^b_{b,k} + \kappa \omega_{\mathrm{corr},k}\right]^\times$$

where $\kappa > 0$ is a correction gain and $\omega_{\mathrm{corr}}$ is a correction angular velocity to align the accelerometer reading $a_b$ with its gravity bias

$$\omega_{\mathrm{corr},k} = \frac{a^b_{b,k}}{||a^b_{b,k}||} \times \left(\hat{R}^o_{b,k}\right)^{\mathrm{T}} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

# 2nd-Stage: Kalman Filter




**Process Model**

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$

$$w_k \sim (0, Q_k)$$

$$p^o_{b,k} = p^o_{b,k-1} + v^o_{b,k-1}\Delta t + \frac{1}{2}\left(\hat{R}^o_{b,k} a^b_{b,k} + a^o_g\right)\Delta t^2$$

$$v^o_{b,k} = v^o_{b,k-1} + \left(\hat{R}^o_{b,k} a^b_{b,k} + a^o_g\right)\Delta t$$

$$p_{i,k} = p_{i,k-1}$$

**Measurement Model**

$$y_k = Cx_k + v_k$$

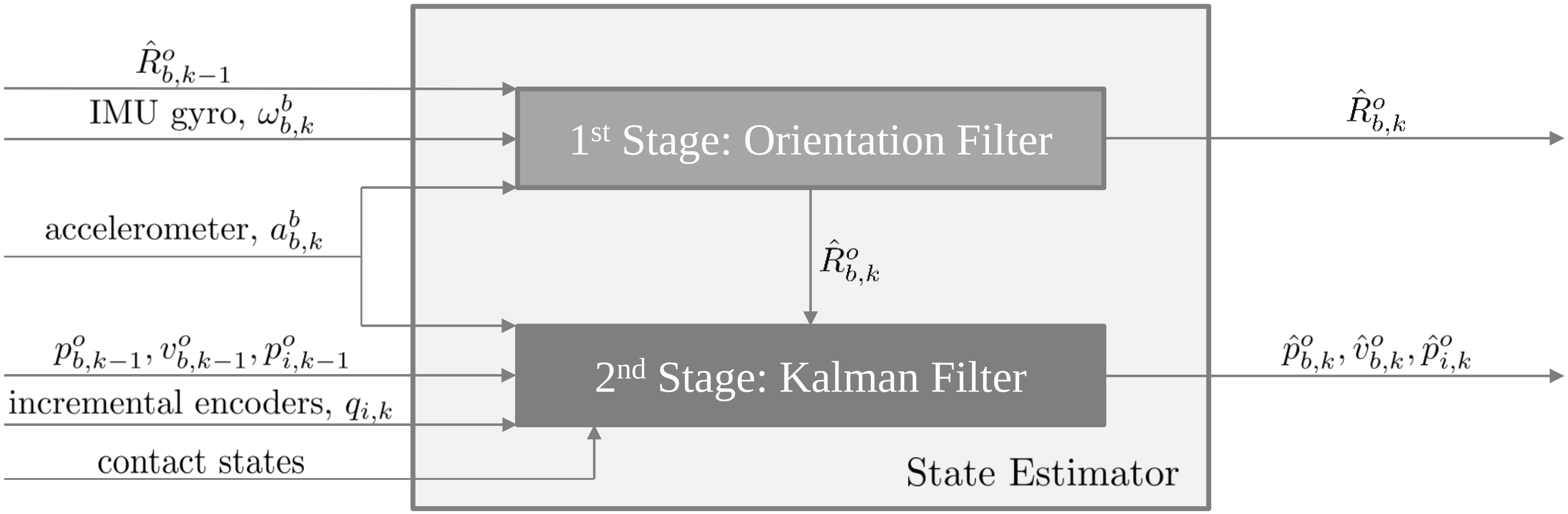
$$v_k \sim (0, R_k)$$

$$p^o_{\text{rel},k}\left(q_i, \hat{R}^o_{b,k}\right) = p^o_{b,k} - p^o_{i,k}$$

$$\dot{p}^o_{\text{rel},k}\left(q_i, \dot{q}_i, \hat{R}^o_{b,k}, \omega^b_{b,k}\right) = v^o_{b,k}$$

$$h_{i,k} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} p^o_{i,k}$$

# $2^{nd}$-Stage: Kalman Filter




**Prediction**

$$\hat{x}_k^- = A\hat{x}_{k-1}^+ + Bu_{k-1}$$
$$P_k^- = AP_{k-1}^+ A^{\mathrm{T}} + Q_{k-1}$$

**Measurement Update**

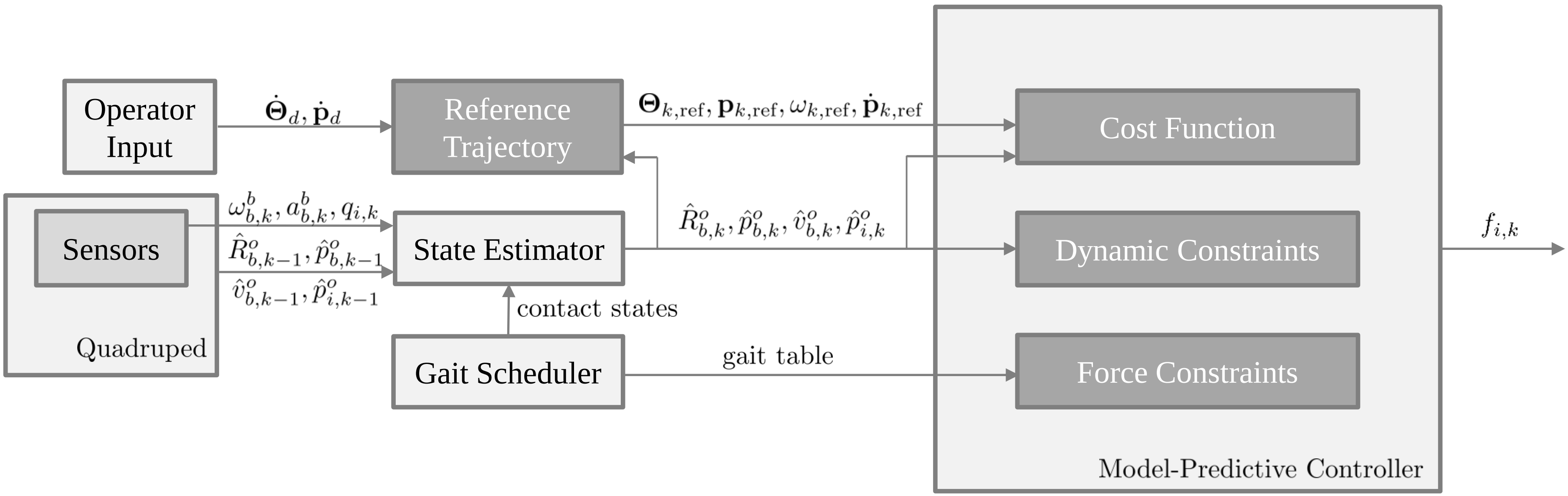
$$K_k = P_k^- C^{\mathrm{T}} \left( C P_k^- C^{\mathrm{T}} + R_k \right)^{-1}$$
$$\hat{x}_k^+ = \hat{x}_k^- + K_k \left( y_k - C\hat{x}_k^- \right)$$
$$P_k^+ = \left( I - K_k C \right) P_k^-$$

**EKF Approach:**

Bloesch M, Hutter M, Hoepflinger M A, et al. State estimation for legged robotsconsistent fusion of leg kinematics and imu[J]. Robotics, 2013, 17: 17-24.

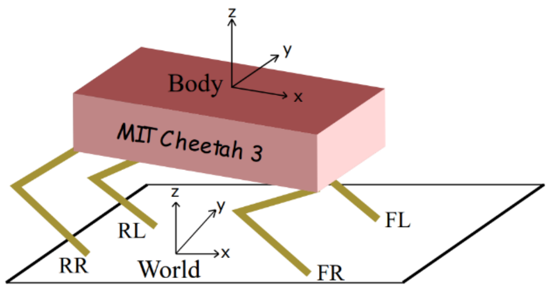
# **Part III:** Body Control: Model-Predictive Controller

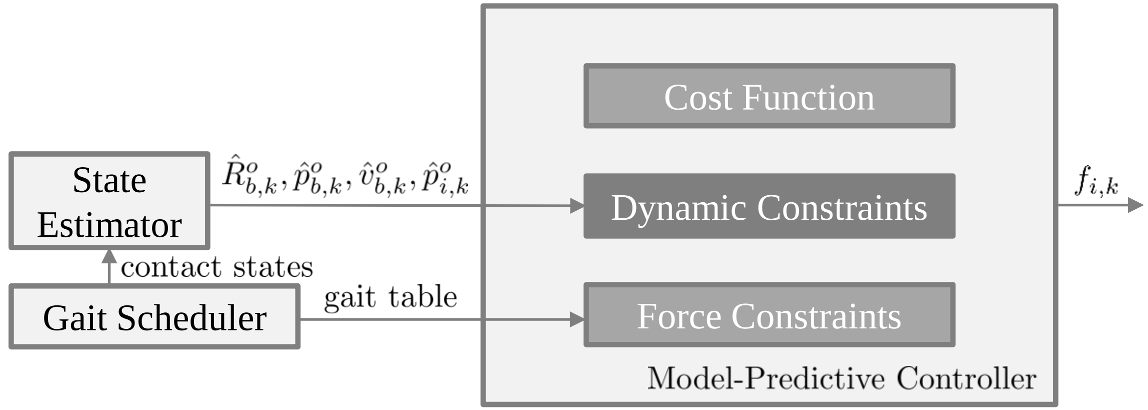
# Framework



Operator Input
$\dot{\mathbf{\Theta}}_d, \dot{\mathbf{p}}_d$
Reference Trajectory
$\mathbf{\Theta}_{k,\mathrm{ref}}, \mathbf{p}_{k,\mathrm{ref}}, \omega_{k,\mathrm{ref}}, \dot{\mathbf{p}}_{k,\mathrm{ref}}$
Cost Function
Sensors
$\omega^b_{b,k}, a^b_{b,k}, q_{i,k}$
$\hat{R}^o_{b,k-1}, \hat{p}^o_{b,k-1}$
$\hat{v}^o_{b,k-1}, \hat{p}^o_{i,k-1}$
State Estimator
$\hat{R}^o_{b,k}, \hat{p}^o_{b,k}, \hat{v}^o_{b,k}, \hat{p}^o_{i,k}$
Dynamic Constraints
$f_{i,k}$
contact states
Quadruped
Gait Scheduler
gait table
Force Constraints
Model-Predictive Controller

# Dynamic Constraints: State Space Model




**Simplified Robot Dynamics**

$$m\ddot{\mathbf{p}} + m\mathbf{g} = \sum_{i=1}^{4} \mathbf{f}_i$$

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathbf{I}_w \omega) = \mathbf{I}_w \dot{\omega} + [\omega]\mathbf{I}_w \omega = \sum_{i=1}^{4} \mathbf{r}_i \times \mathbf{f}_i$$

$$\mathbf{I}_w = \mathbf{R}\mathbf{I}_B\mathbf{R}^{\mathrm{T}}$$

$$\omega = \mathbf{J}_\omega \dot{\mathbf{\Theta}}$$

**Continuous Time Model**

$$\dot{\mathbf{x}}(t) = \mathbf{A}_c(\mathbf{\Theta})\mathbf{x}(t) + \mathbf{B}_c(\mathbf{p}, \mathbf{\Theta})\mathbf{u}(t)$$

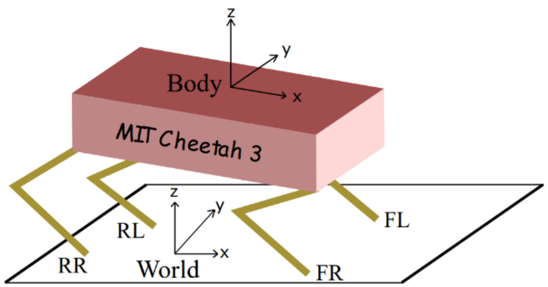
**Discrete Time Model**

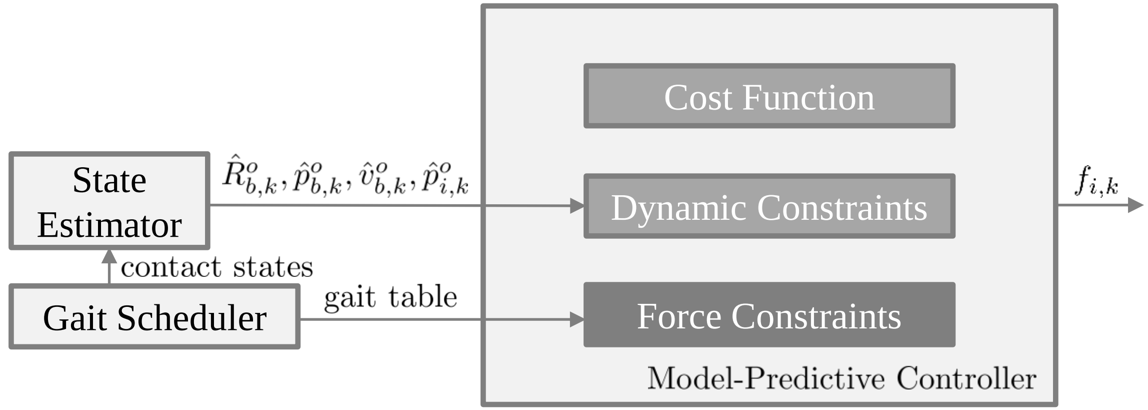
$$\exp\left(\begin{bmatrix} \mathbf{A}_c & \mathbf{B}_c \\ 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{A}_d & \mathbf{B}_d \\ 0 & \mathbf{I} \end{bmatrix}$$

**State Space Model:** add an additional gravity state.

$$\begin{bmatrix} \dot{\mathbf{\Theta}} \\ \dot{\mathbf{p}} \\ \dot{\omega} \\ \ddot{\mathbf{p}} \\ \dot{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{J}_\omega^{-1} & \mathbf{0}_{33} & \mathbf{0}_{31} \\ \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{I}_3 & \mathbf{0}_{31} \\ \mathbf{0}_{33} & \mathbf{0}_{33} & -\mathbf{I}_w^{-1}[\omega]\mathbf{I}_w & \mathbf{0}_{33} & \mathbf{0}_{31} \\ \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{e}_z \\ \mathbf{0}_{13} & \mathbf{0}_{13} & \mathbf{0}_{13} & \mathbf{0}_{13} & \mathbf{0}_{11} \end{bmatrix} \begin{bmatrix} \mathbf{\Theta} \\ \mathbf{p} \\ \omega \\ \dot{\mathbf{p}} \\ \mathbf{g} \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{0}_{33} \\ \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{0}_{33} & \mathbf{0}_{33} \\ \mathbf{I}_w^{-1}[\mathbf{r}_1] & \mathbf{I}_w^{-1}[\mathbf{r}_2] & \mathbf{I}_w^{-1}[\mathbf{r}_3] & \mathbf{I}_w^{-1}[\mathbf{r}_4] \\ \frac{\mathbf{I}_3}{m} & \frac{\mathbf{I}_3}{m} & \frac{\mathbf{I}_3}{m} & \frac{\mathbf{I}_3}{m} \\ \mathbf{0}_{13} & \mathbf{0}_{13} & \mathbf{0}_{13} & \mathbf{0}_{13} \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \\ \mathbf{f}_4 \end{bmatrix}$$

# Force Constraints



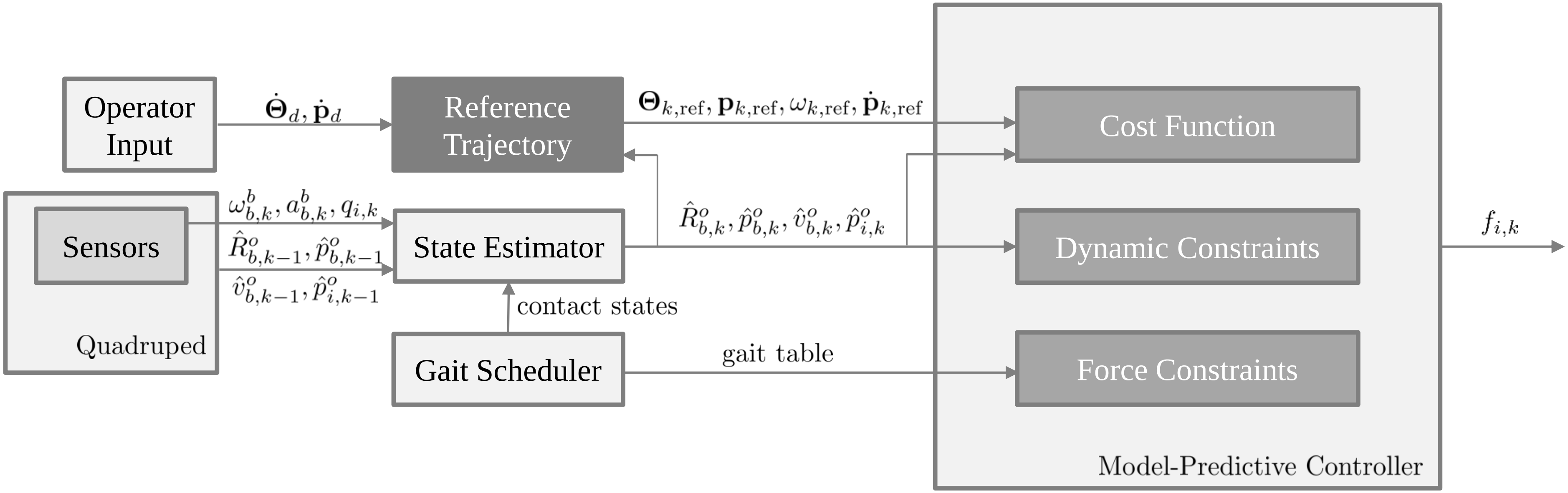




**Equality Constraint:** $\mathbf{D}_k \mathbf{u}_k = 0.$
used to set all forces from feet off the ground to 0, enforcing the desired gait.

**Inequality Constraint:** $\underline{\mathbf{c}}_k \leq \mathbf{C}_k \mathbf{u}_k \leq \overline{\mathbf{c}}_k.$
used to limit the minimum and maximum $z$-force as well as a square pyramid approximation of the friction cone for each foot on the ground.

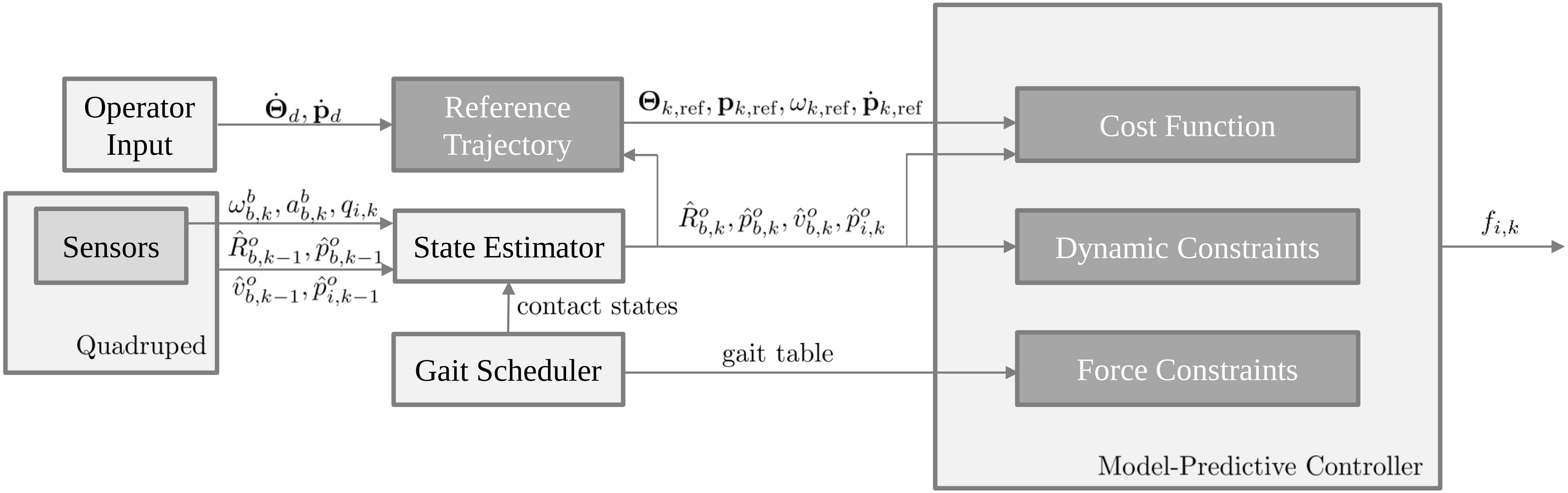
$$\begin{bmatrix} 0 \\ -\infty \\ 0 \\ -\infty \\ f_{\min} \end{bmatrix} \leq \begin{bmatrix} 1 & 0 & \mu \\ 1 & 0 & -\mu \\ 0 & 1 & \mu \\ 0 & 1 & -\mu \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{ix} \\ f_{iy} \\ f_{iz} \end{bmatrix} \leq \begin{bmatrix} \infty \\ 0 \\ \infty \\ 0 \\ f_{\max} \end{bmatrix} \Rightarrow \begin{bmatrix} \underline{\mathbf{c}}_1 \\ \underline{\mathbf{c}}_2 \\ \underline{\mathbf{c}}_3 \\ \underline{\mathbf{c}}_4 \end{bmatrix} \leq \begin{bmatrix} \mathbf{C}_1 & & & \\ & \mathbf{C}_2 & & \\ & & \mathbf{C}_3 & \\ & & & \mathbf{C}_4 \end{bmatrix} \mathbf{U} \leq \begin{bmatrix} \overline{\mathbf{c}}_1 \\ \overline{\mathbf{c}}_2 \\ \overline{\mathbf{c}}_3 \\ \overline{\mathbf{c}}_4 \end{bmatrix}$$

# Reference CoM Trajectory Generation




- The reference trajectories are simple and only contain non-zero $xy$-velocity, $xy$-position, $z$-position, yaw and yaw rate.
- All parameters are commanded directly by the robot operator except for yaw and $xy$-position, which are determined by integrating the appropriate velocities.
- The other states are always set to 0.
- The reference trajectory is also used to determine the dynamics constraints and future foot placement locations.

# MPC Formulation




$$\min_{\mathbf{x},\mathbf{u}} \sum_{k=0}^{j-1} ||\mathbf{x}_{k+1} - \mathbf{x}_{k+1,\mathrm{ref}}||_{\mathbf{Q}_k} + ||\mathbf{u}_k||_{\mathbf{R}_k}$$

$$\text{s.t.} \quad \mathbf{x}_{k+1} = \mathbf{A}_k \mathbf{x}_k + \mathbf{B}_k \mathbf{u}_k$$

$$\underline{\mathbf{c}}_k \le \mathbf{C}_k \mathbf{u}_k \le \overline{\mathbf{c}}_k$$

$$\mathbf{D}_k \mathbf{u}_k = 0$$

$$\min_{\mathbf{U}} \frac{1}{2}\mathbf{U}^{\mathbf{T}}\mathbf{H}\mathbf{U} + \mathbf{U}^{\mathbf{T}}\mathbf{g}$$

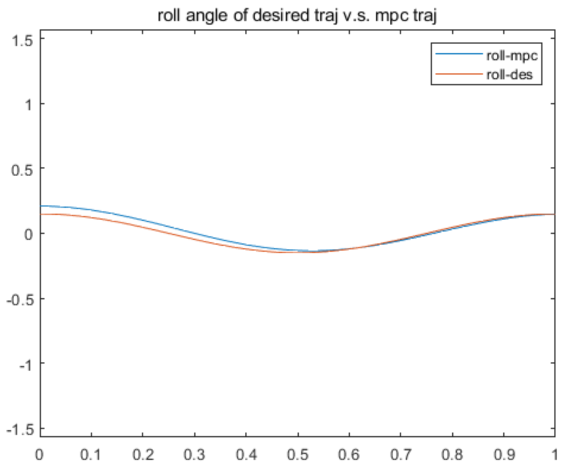
$$\text{s.t.} \quad \underline{\mathbf{c}} \le \mathbf{C}\mathbf{U} \le \overline{\mathbf{c}}$$

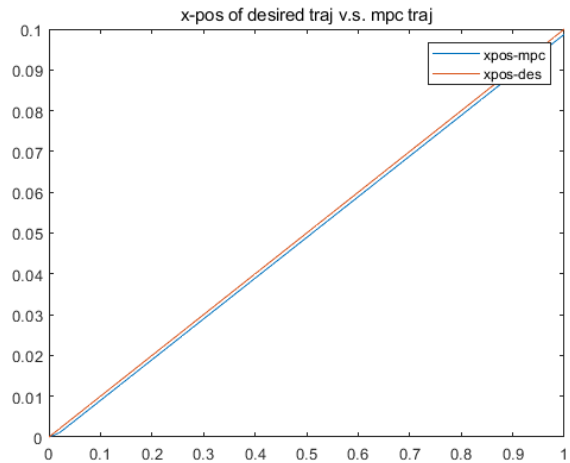
Controller Settings and Robot Data

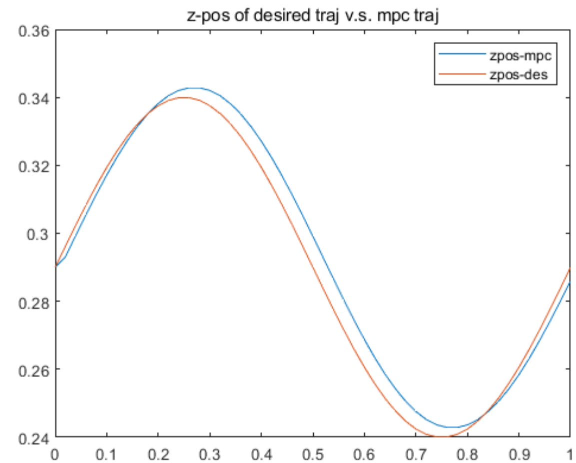
| | |
|---|---|
| m | 9.0 kg |
| $I_{xx}$ | 0.025 kg·m$^2$ |
| $I_{yy}$ | 2.1 kg·m$^2$ |
| $I_{zz}$ | 2.1 kg·m$^2$ |
| $\mu$ | 0.6 |
| $g_z$ | -9.8 m/s$^2$ |
| $\tau_{\max}$ | 250 N·m |

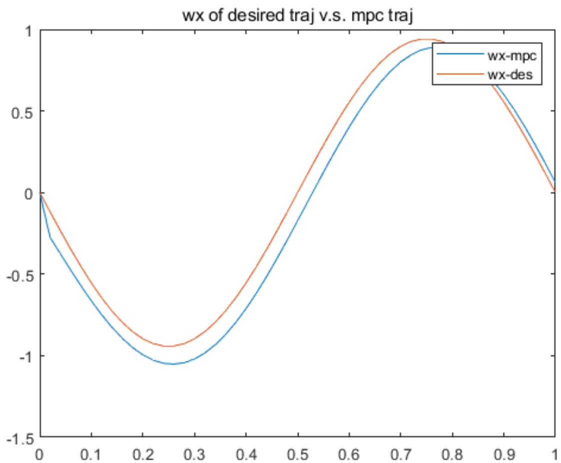
| | |
|---|---|
| $\boldsymbol{\Theta}$ weight | 30 |
| $\mathbf{p}$ weight | 50 |
| $\omega$ weight | 1 |
| $\dot{\mathbf{p}}$ weight | 50 |
| $\mathbf{f}$ weight | 1e-6 |
| $f_{\min}$ | 5 |
| $f_{\max}$ | 150 |

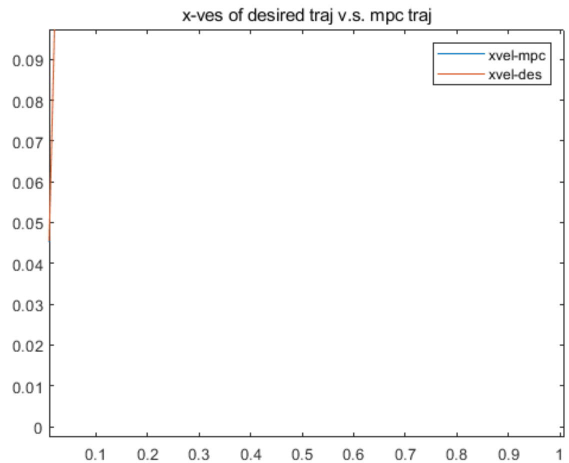
# Simulation Results

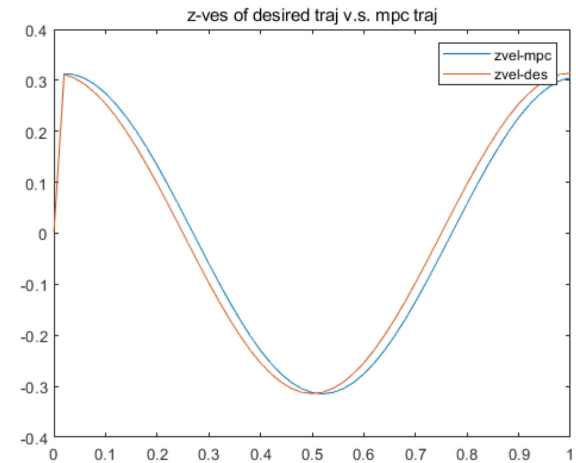


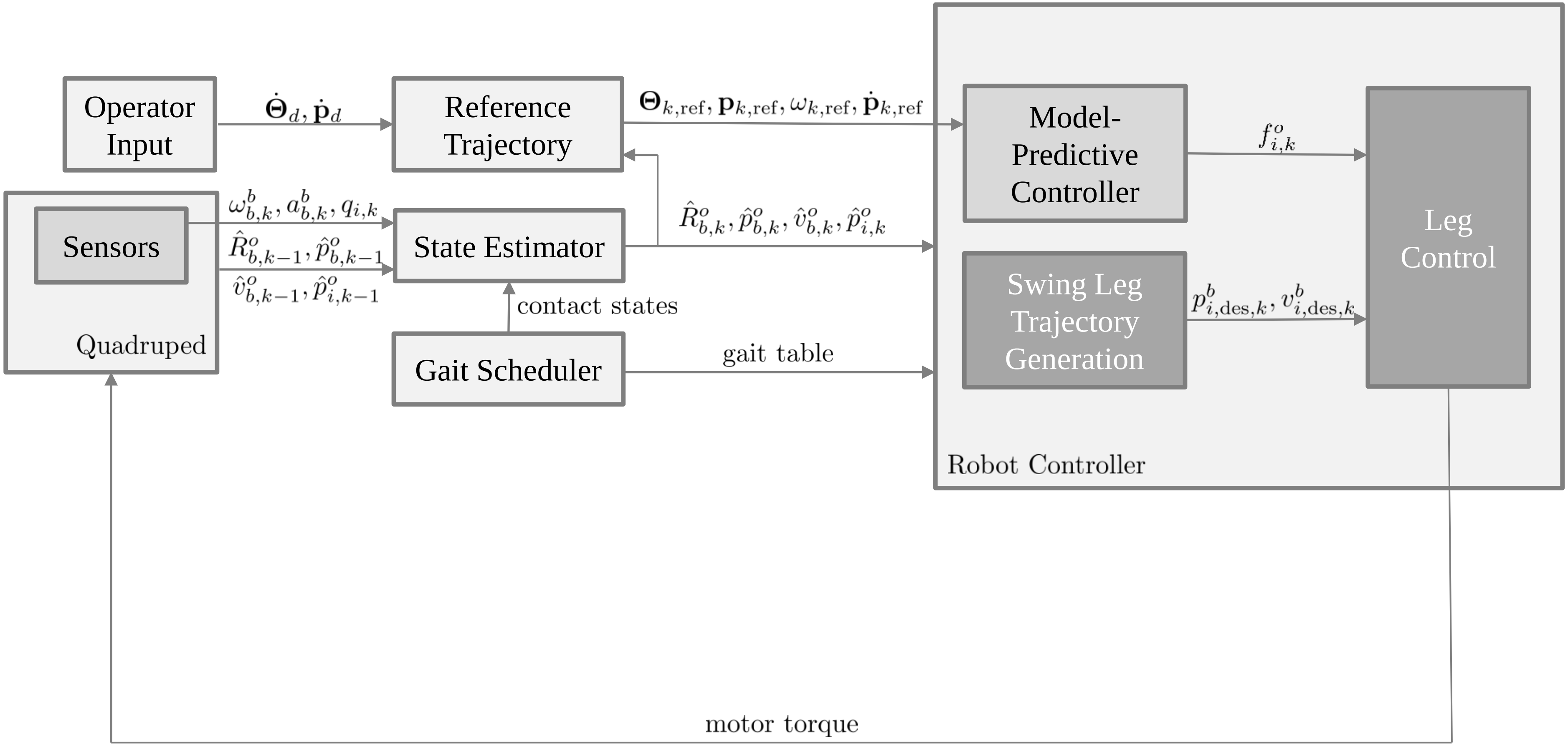



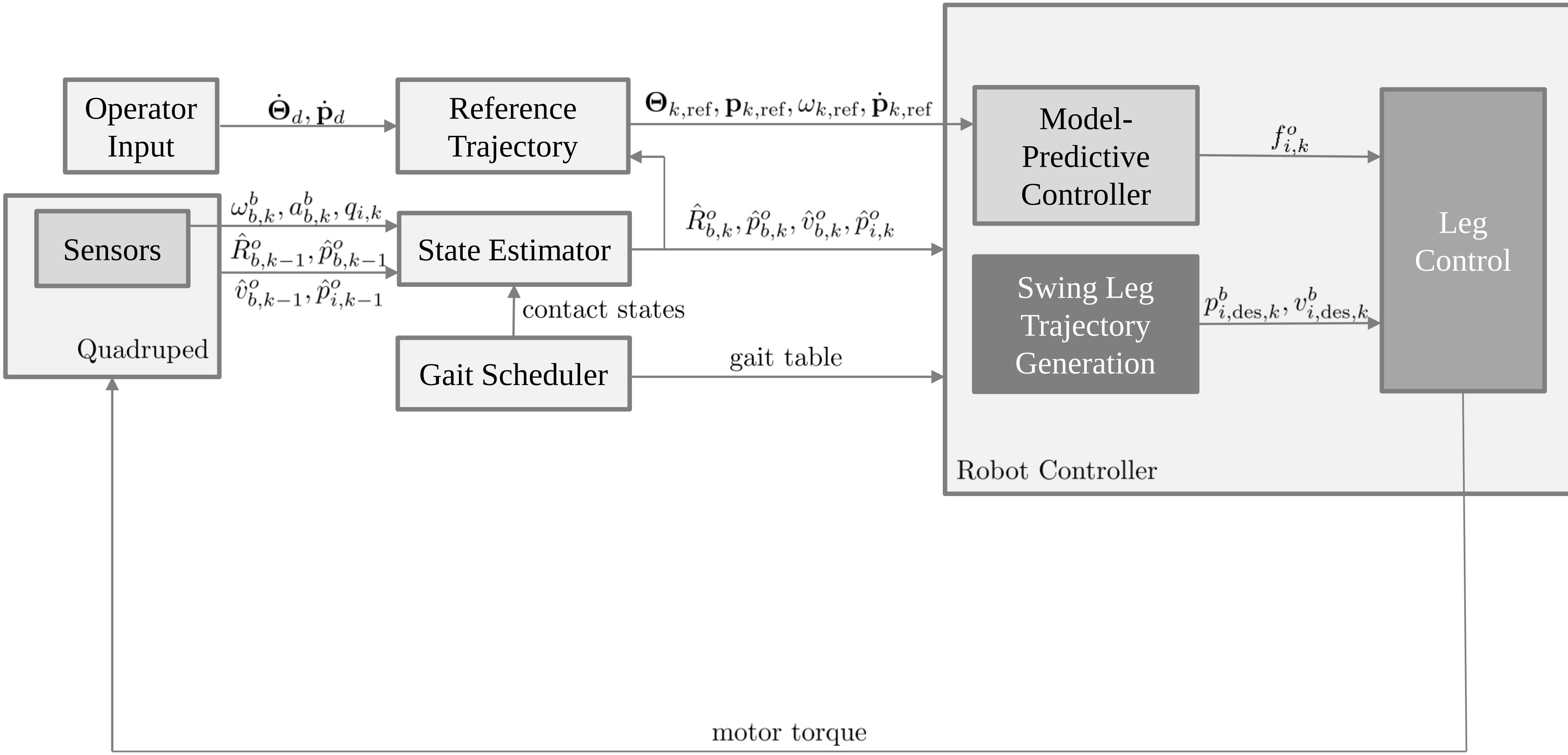
# **Part IV:** Swing Leg Trajectory and Leg Control

# Framework



Operator Input
$\dot{\mathbf{\Theta}}_d, \dot{\mathbf{p}}_d$
Reference Trajectory
$\mathbf{\Theta}_{k,\mathrm{ref}}, \mathbf{p}_{k,\mathrm{ref}}, \omega_{k,\mathrm{ref}}, \dot{\mathbf{p}}_{k,\mathrm{ref}}$
Model-Predictive Controller
$f^o_{i,k}$
Leg Control
Sensors
Quadruped
$\omega^b_{b,k}, a^b_{b,k}, q_{i,k}$
$\hat{R}^o_{b,k-1}, \hat{p}^o_{b,k-1}$
$\hat{v}^o_{b,k-1}, \hat{p}^o_{i,k-1}$
State Estimator
$\hat{R}^o_{b,k}, \hat{p}^o_{b,k}, \hat{v}^o_{b,k}, \hat{p}^o_{i,k}$
Swing Leg Trajectory Generation
$p^b_{i,\mathrm{des},k}, v^b_{i,\mathrm{des},k}$
contact states
Gait Scheduler
gait table
Robot Controller
motor torque

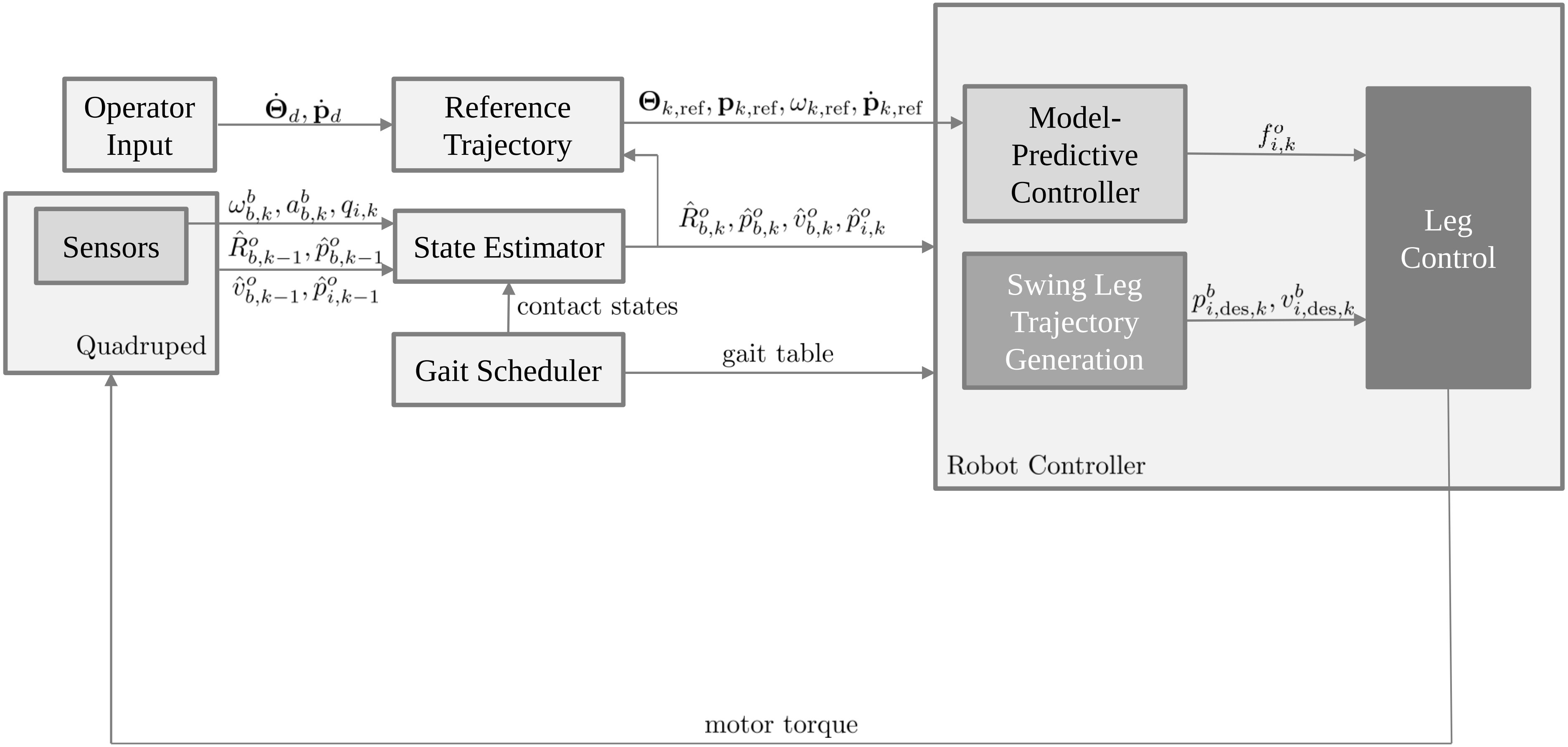
# Swing Leg Trajectory Generation




**Desired Foot-Position Trajectory:** $p^b_{i,\mathrm{des},k} = (R^o_{b,k})^{\mathrm{T}} \left( p^o_{i,\mathrm{des},k} - \hat{p}^o_{b,k} \right)$

**Desired Foot-Velocity Trajectory:** $v^b_{i,\mathrm{des},k} = (R^o_{b,k})^{\mathrm{T}} \left( v^o_{i,\mathrm{des},k} - \hat{v}^o_{b,k} \right)$

# Leg Control

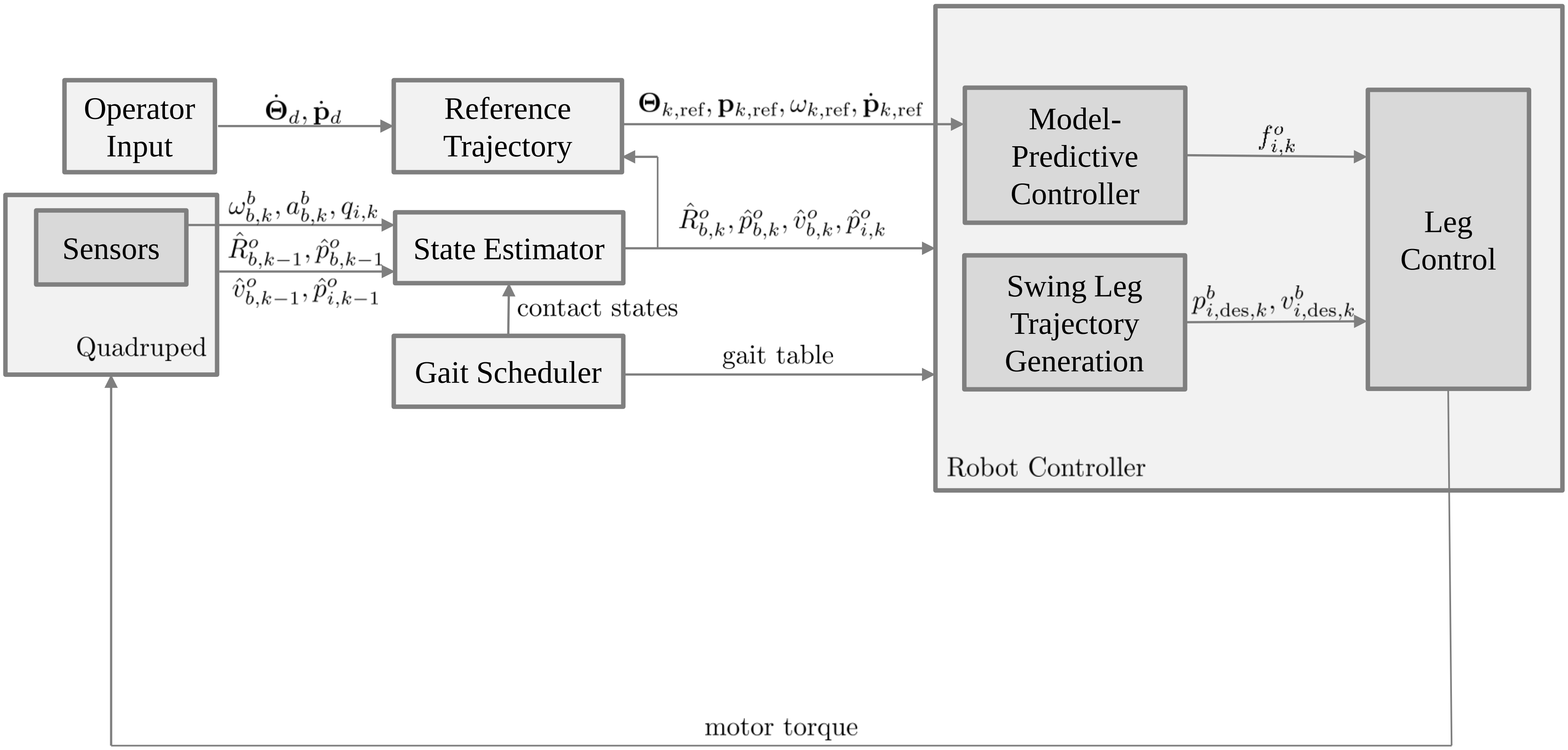



$$\textbf{Swing Leg Control: } \tau_{i,k} = J^{\mathrm{T}}_{i,k}\left[K_p\left(p_{i,\mathrm{des},k} - \hat{p}_{i,k}\right) + K_d\left(v_{i,\mathrm{des},k} - \hat{v}_{i,k}\right)\right]$$

$$\textbf{Stance Leg Control: } \tau_{i,k} = J^{\mathrm{T}}_{i,k} f_{i,k}$$

## **Part V:** Summary

# The Control Architecture for MIT Cheetah 3



# References

[1] Bledt G, Powell M J, Katz B, et al. MIT Cheetah 3: Design and control of a robust, dynamic quadruped robot[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 2245-2252.

[2] Bloesch M, Hutter M, Hoepflinger M A, et al. State estimation for legged robotsconsistent fusion of leg kinematics and imu[J]. Robotics, 2013, 17: 17-24.

[3] Mahony R, Hamel T, Pflimlin J M. Nonlinear complementary filters on the special orthogonal group[J]. IEEE Transactions on automatic control, 2008, 53(5): 1203-1218.

[4] Di Carlo J, Wensing P M, Katz B, et al. Dynamic locomotion in the mit cheetah 3 through convex model-predictive control[C]//2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2018: 1-9.

# References

[5] Jerez J L, Kerrigan E C, Constantinides G A. A condensed and sparse QP formulation for predictive control[C]//2011 50th IEEE Conference on Decision and Control and European Control Conference. IEEE, 2011: 5217-5222.

[6] Ferreau H J, Kirches C, Potschka A, et al. qpOASES: A parametric active-set algorithm for quadratic programming[J]. Mathematical Programming Computation, 2014, 6(4): 327-363.

[7] Guennebaud G, Jacob B. Eigen[J]. URl: http://eigen. tuxfamily. org, 2010, 3.

# Thanks for Listening