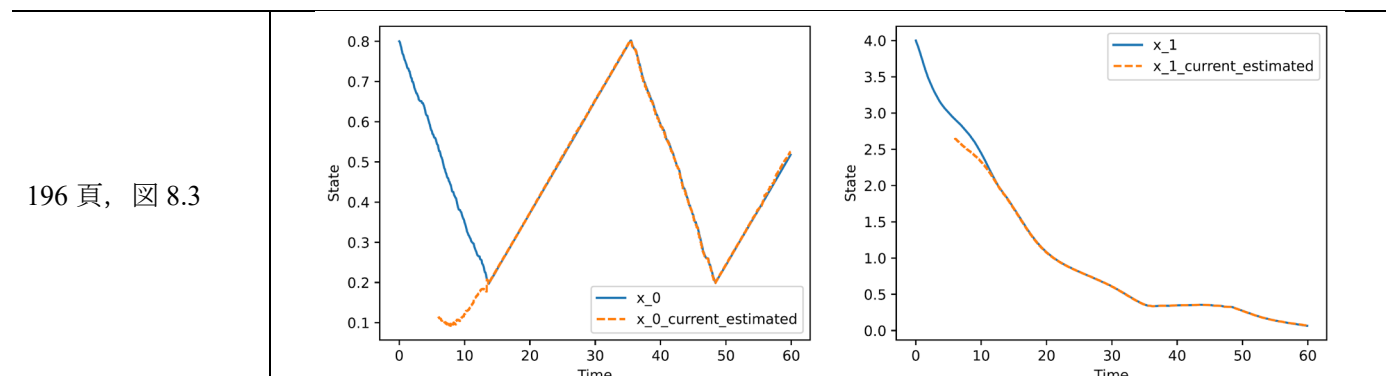


## 『Python と CasADi で学ぶモデル予測制御』正誤表と補足事項（第1刷用）

正誤箇所	誤	正
48 頁, 下 6 行目	ubx=決定変数の下 <del>下</del> 限値を表すリスト,	ubx=決定変数の上 <del>上</del> 限値を表すリスト,
49 頁, 6 行目	その場合, 省略した引数には 0 または 0 のみを要素に持つ適切な大きさのリストが引数に渡されたときと同じふるまいをします. S の引数のふるまいに関する詳細は 3.6 節も参照してください.	その場合, 決定変数の初期値は 0, 決定変数, 制約条件の上限は $+\infty$ , 下限は $-\infty$ が与えられたときと同じふるまいをします.
83 頁, 3 行目	上 <del>上</del> 限値 lbx を作成しています	下 <del>下</del> 限値 lbx を作成しています
87 頁, 式(5.2)の 3 行目第 1 項	$\frac{1}{2}\dot{x}^2$	$\frac{1}{2}(M + m)\dot{x}^2$
90 頁, 下 4 行目 90 頁. 式(5.14)中 91 頁, 式(5.15)中	$k_s + T$	$t_s + T$
93 頁, 5 行目	状態変数を目標値に安定化させるような入力制御は	状態変数が目標値のときに平衡点であるような入力制御は
95 頁, プログラム 22,23 行目	nu = 1 # 状態変数の次元 nx = 4 # 制御入力の次元	nu = 1 # 制御入力の次元 nx = 4 # 状態変数の次元
98 頁, 5.6.7 項のプログラム 4,5 行目	lbx = x_init + x_lb*K + u_lb*K ubx = x_init + x_ub*K + u_ub*K	lbx = x_init + x_lb*K + u_lb*K ubx = x_init + x_ub*K + u_ub*K
105 頁, 下 2 行目	$\lambda(T) = \frac{\partial}{\partial x} \phi(\mathbf{x}(T))$	$\lambda(T) = \frac{\partial}{\partial x} \phi(\mathbf{x}(T), T)$
142 頁, 下 4 行目	制御入力 $\mathbf{u}_{\text{ref}}$ は	制御入力の目標値 $\mathbf{u}_{\text{ref}}$ は
194 頁, 8.4.7 項のプログラム	10 行目「x0 = make_x0(x_hat)」を削除	
195 頁, 8.4.7 項のプログラム	19 行目「x_current = x_init」下に同じインデントで右記を追加	X_current_est = []
195 頁, 8.4.7 項のプログラム	27 行目「X_est.append(x_hat)」下に同じインデントで右記を追加	X_current_est.append(x0[nx*K:nx*(K+1)])
196 頁, プログラム	4 行目の下に同じインデントで右記を追加	X_current_est = np.array(X_current_est).reshape(-1, nx)
196 頁, プログラム 10 行目	t_eval[:, -K], X[:, -K, i]	t_eval[:, :, X[:, i]
196 頁, プログラム 11 行目	t_eval[:, -K], X_est[:, i]	t_eval[K:], X_current_est[:, i]
196 頁, プログラム 12 行目	"x_{i}_estimated"	"x_{i}_current_estimated"



## 補足事項

51 頁 問題について	<p>ここで扱う問題は下記の文献を参考にしています。  M. Diehl, et al. Linear and nonlinear optimization with CasADi.  <a href="https://www.syscop.de/files/users/Adrian.buerger/exercise_sheet_numerical_optimization_for_practitioners_with_solutions.pdf">https://www.syscop.de/files/users/Adrian.buerger/exercise_sheet_numerical_optimization_for_practitioners_with_solutions.pdf</a></p>
60 頁 @1 について	<p>@1 とあるのは, SX 型の要素に共通して現れる部分式を表す記号です. CasADi がこのような形で共通して現れる部分式を扱っていることは, MPC を効率的に解くうえで重要な要素となっています. (文献[26]参照)</p>
71, 72 頁 式(4.4)中の $R$ について	<p><math>R</math> は正の実数と同一視できますが, ここではあえて行列として扱いたいためこのような表現になっています.</p>
4 章以降の MPC の実装に際して	<p>MPC の実装では, 以下のモジュール群をインポートしています (第 4 章～付録 B で共通). 詳しくはサポートサイトのソースコードをご参照ください.</p> <pre>import time import casadi import numpy as np import matplotlib.pyplot as plt import matplotlib.patches as patches from matplotlib.animation import FuncAnimation</pre>
231 頁 付録 A.4 の基本的なアイデア 3 点について	<p>書籍では概要のみを説明し, サポートサイトで数式とあわせて説明をしています.</p>
262 頁 Pinocchio のバージョンについて	<p>本書では, バージョン 3.0.0 がインストールされた場合の動作を確認しています.</p>