基于机器学习的信息物理系统安全控制

刘 坤 [1] 马书鹤 [1] 马奥运 [1] 张淇瑞 [1] 夏元清 [1]

摘 要 研究了控制信号被恶意篡改的信息物理系统的安全控制问题. 首先, 提出一种改进果蝇优化核极限学习机算法 (Kernel extreme learning machine with improved fruit fly optimization algorithm, IFOA-KELM) 对攻击信号进行重构. 然后, 将所得重构信号作为系统扰动加以补偿, 进而设计模型预测控制策略, 并给出了使被控系统是输入到状态稳定的条件. 另外, 本文从攻击者角度建立优化模型得到最优攻击策略用以生成足够的受攻击数据, 基于此数据, 来训练改进果蝇优化核极限学习机算法. 最后, 使用弹簧–质量–阻尼系统进行仿真, 验证了改进果蝇优化极限学习机算法和所提安全控制策略的有效性.

关键词 信息物理系统, 攻击信号重构, 核极限学习机, 果蝇优化算法, 模型预测控制

引用格式 刘坤, 马书鹤, 马奥运, 张淇瑞, 夏元清. 基于机器学习的信息物理系统安全控制. 自动化学报, 2021, **47**(6): 1273−1283

DOI 10.16383/j.aas.c190352

Secure Control for Cyber-physical Systems Based on Machine Learning

LIU Kun[1] MA Shu-He[1] MA Ao-Yun[1] ZHANG Qi-Rui[1] XIA Yuan-Qing[1]

Abstract This paper investigates the security control problem of cyber-physical systems whose control signals are maliciously tampered. Firstly, a kernel extreme learning machine with improved fruit fly optimization (IFOA-KELM) algorithm is proposed to reconstruct the attack signal. Secondly, with the reconstructed signal treated as disturbance, a model predictive control strategy is designed to secure the system, and a condition that guarantees the input-to-state stability of the attacked system is given. In addition, to train the proposed algorithm, enough data of the system attacked with an optimal strategy is generated. This strategy is obtained by solving an optimization problem from the attacker′s perspective. Finally, a numerical example of the spring-mass-damping system is illustrated to verify the effectiveness of the IFOA-KELM algorithm and the proposed control strategy.

Key words Cyber-physical systems, attack signal reconstruction, kernel extreme learning machine (KELM), fruit fly optimization algorithm (FOA), model predictive control

Citation Liu Kun, Ma Shu-He, Ma Ao-Yun, Zhang Qi-Rui, Xia Yuan-Qing. Secure control for cyber-physical systems based on machine learning. *Acta Automatica Sinica*, 2021, **47**(6): 1273−1283

信息物理系统 (Cyber-physical systems, CPSs) 是计算单元与物理对象在网络空间中高度集成交互形成的智能系统[1−2]. CPSs 广泛应用于水净化与分配[3−4]、智能电网[5]、智能交通[6] 和国防军事[7] 等重要领域. 然而, 网络的开放性使得 CPSs 极易受到攻击, 这对人们的经济和生活产生了巨大危害[8−9]. 如: 2019 年 3 月全球最大铝生产商挪威海德鲁公司的勒索病毒攻击事件, 2019 年 1 月委内瑞拉水电站的网络攻击事件, 2017 年美国制药公司默克的勒索病毒攻击事件, 2014 年美国波士顿儿童医院的大规模分布式拒绝服务攻击事件等. 因此, 研究 CPSs 安全相关的理论和技术刻不容缓.

常见的研究 CPSs 安全问题的方法有: Lyapunov 方法[10−11]、最优化方法[12−13]、博弈论方法[14] 等. 近年来, 人工智能、云计算等技术的发展, 为解决 CPSs 的安全问题提供了新的途径和方法. 不过, 值得注意的是, 现有的大部分研究成果主要着重于攻击的检测和识别, 如: Vu 等[15] 利用 K-最近邻算法对网络状态进行分类以主动检测分布式拒绝服务攻

收稿日期 2019-05-10 录用日期 2019-10-21
Manuscript received May 10, 2019; accepted October 21, 2019
国家自然科学基金 (61873034, 61503026, 61836001), 北京自然科学基金 (4182057), 国家自然科学基金重大国际 (地区) 合作项目 (61720106010), 北京市智能物流系统协同创新中心开放课题 (BILSCIC-2019KF-13), 北京理工大学研究生创新项目 (2019CX20031) 资助
Supported by National Natural Science Foundation of China (61873034, 61503026, 61836001), Beijing Natural Science Foundation (4182057), Major International (Regional) Joint Research Project of National Natural Science Foundation of China (61720106010), the Open Subject of Beijing Intelligent Logistics System Collaborative Innovation Center (BILSCIC-2019KF-13), and Graduate Technological Innovation Project of Beijing Institute of Technology (2019CX20031)
本文责任编委 曹向辉
Recommended by Associate Editor CAO Xiang-Hui
1. 北京理工大学自动化学院复杂系统智能控制与决策国家重点实验室 北京 100081
1. Key Laboratory of Intelligent Control and Decision of Complex Systems, School of Automation, Beijing Institute of Technology, Beijing 100081

击; Kumar 和 Devaraj[16] 先采用基于互信息的特征选择方法选取网络的重要特征, 再将它们作为反向传播神经网络 (Back-propagation neural network, BPNN) 的输入用以识别系统中各种类型的入侵事件; Nawaz 等[17] 和 Esmalifalak 等[18] 利用支持向量机 (Support vector machine, SVM) 算法检测智能电网中的虚假数据注入攻击; Kiss 等[19] 利用高斯混合模型算法对田纳西–伊士曼过程中的传感器测量值进行聚类, 并选取轮廓系数作为评价指标有效识别攻击; Inoue 等[3] 基于由 SWaT 系统产生的时间序列数据 (包括正常和攻击数据) 对比了深度神经网络 (Deep neural network, DNN) 和支持向量机 (SVM) 两种算法的攻击检测效果, 整体而言 DNN 要优于 SVM. 然而, 对于某些情况仅仅做到攻击检测与识别是不够的, 还需要考虑对攻击信号进行重构进而设计出合适的安全控制器, 以削弱甚至消除攻击对系统造成的影响和危害.

本文利用机器学习技术对攻击信号进行重构, 本质上是对从受攻击 CPS 中采集到的数据进行拟合回归进而获取攻击策略的过程. 常见的机器学习回归方法有: BPNN、高斯过程 (Gaussian process, GP)、极限学习机 (Exteme learning machine, ELM)、最小二乘支持向量机 (Least square support vector machine, LS-SVM) 等. 文献 [20] 为了提高非线性系统控制器的控制精度, 分别利用 BPNN、ELM 和 LS-SVM 对系统的未建模动态部分以及线性化误差进行精确估计和补偿, 并从算法的训练时长和测试误差角度对三种算法进行了对比, 结果表明训练 ELM 用时最短, LS-SVM 拟合精度最高. 为了改善上述算法各自存在的弊端, 在原始算法基础上出现了不同的变体, 如: 为了改善 BPNN 存在的训练速度慢、参数寻优难、过拟合、局部最优以及隐含层节点数人为指定等问题, 文献 [21] 利用引入了自适应学习率和动量项的粒子群优化 BP 神经网络 (Particle swarm optimization BP neural network, PSO-BP) 预测网络流量; 为了提高 ELM 的稳健性和非线性逼近能力, Huang 等[22] 提出了核极限学习机 (Kernel extreme learning machine, KELM), 并通过实验验证了 KELM 比 LS-SVM 具有更强的泛化能力.

基于以上考虑, 本文利用 KELM 重构攻击信号, 但是考虑到 KELM 同样具有参数敏感性问题, 于是选择具有结构简单、调整参数少、计算量小、收敛速度快等优点的果蝇优化算法 (Fruit fly optimization algorithm, FOA) 对 KELM 进行优化. 然而, 基础的 FOA 存在容易陷入局部最优解的问题, 因此本文对 FOA 进行改进, 最终利用基于改进的果蝇优化算法 (Improved fruit fly optimization algorithm, IFOA) 的 KELM 对攻击信号进行重构, 将攻击信号视作系统扰动并利用重构的攻击信号对其进行补偿, 对补偿后的系统使用模型预测控制 (Model predictive control, MPC) 策略, 并给出了使系统是输入到状态稳定 (Input-to-state stable, ISS) 的条件. 此外, 为了验证所提算法的有效性, 本文从攻击者角度建立了优化模型用以生成攻击数据. 最终, 通过数值仿真验证了 IFOA-KELM 相较于 FOA-KELM、PSO-BP 和 LS-SVM 的优越性以及安全控制策略的有效性.

符号说明. $\mathbf{R}^n$ 表示 n 维欧几里得空间; I_n 表示 n 阶单位阵; $A > 0$ ($A \geq 0$) 表示矩阵 A 是正定矩阵 (半正定矩阵); 对于列向量 $\boldsymbol{x}$ 和矩阵 $P > 0$, $\|\boldsymbol{x}\|$ 表示 $\boldsymbol{x}$ 的 2 范数, $\|\boldsymbol{x}\|_P = \sqrt{\boldsymbol{x}^{\mathrm{T}} P \boldsymbol{x}}$ 表示 $\boldsymbol{x}$ 的加权范数.

1 问题描述

本文考虑执行器受到攻击或控制信号遭受篡改的 CPS, 具体如图 1 所示.

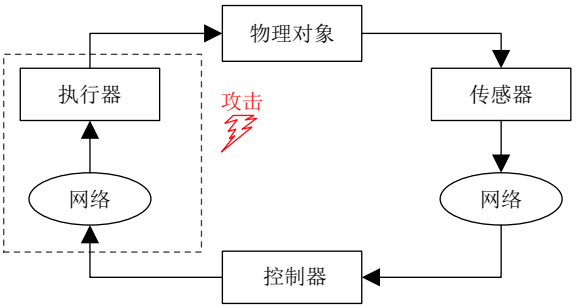


图 1 遭受攻击的信息物理系统框图

Fig. 1 The diagram of the CPS under cyber attack

图 1 中的物理对象是线性时不变系统, 它的状态方程可表示为

$$\boldsymbol{x}(k+1) = A\boldsymbol{x}(k) + B\boldsymbol{u}(k) \tag{1}$$

其中, $\boldsymbol{x}(k) \in \mathbf{R}^n$ 和 $\boldsymbol{u}(k) = K\boldsymbol{x}(k) \in \mathbf{R}^m$ 分别为 k 时刻的系统状态变量和控制输入, A, B 和 K 分别为相应的系统矩阵, 输入矩阵和状态反馈增益矩阵.

假设攻击者根据系统的状态和控制输入设计攻击策略, 则可将受攻击后系统的状态方程表示为

$$\begin{aligned}\boldsymbol{x}_c(k+1) = {} & A\boldsymbol{x}_c(k) + B\boldsymbol{u}_c(k) + B_a\boldsymbol{a}(k) = \\ & A\boldsymbol{x}_c(k) + B\boldsymbol{u}_c(k) + a(\boldsymbol{x}_c(k), \boldsymbol{u}_c(k)) = \\ & A\boldsymbol{x}_c(k) + BK\boldsymbol{x}_c(k) + \\ & a(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))\end{aligned} \tag{2}$$

其中, $\boldsymbol{x}_c(k)$ 和 $\boldsymbol{u}_c(k)$ 分别为受攻击后系统的状态变量和控制输入, 矩阵 B_a 描述了攻击对系统产生的

影响, $\boldsymbol{a}(k)$ 表示攻击信号, 函数 $a(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k)) = B_a\boldsymbol{a}(k)$ 为待设计的攻击者策略.

为了获得足够的标记数据用于训练机器学习算法, 我们从攻击者角度出发, 建立系统的优化模型.

令 $A_1 = A + BK$, 对于攻击者而言, 系统的状态方程可表示为

$$\boldsymbol{x}_c(k+1) = A_1\boldsymbol{x}_c(k) + B_a\boldsymbol{a}(k) \tag{3}$$

攻击者的目标往往是用较少代价使得系统状态尽量偏离其期望的轨迹, 因此可以用如下优化问题来描述攻击者的攻击目标

$$\max J = \sum_{k=0}^{\infty}\left[\boldsymbol{e}^{\mathrm{T}}(k)Q\boldsymbol{e}(k) - \boldsymbol{a}^{\mathrm{T}}(k)R\boldsymbol{a}(k)\right] \tag{4}$$

其中, $\boldsymbol{e}(k) = \boldsymbol{x}_c(k) - \boldsymbol{x}(k)$ 表示系统的状态误差, $Q \geq 0$ 和 $R \geq 0$ 分别为状态误差和攻击信号的加权矩阵.

为了方便求解上述优化问题, 定义如下变量

$$\bar{\boldsymbol{x}}(k) = \begin{bmatrix} \boldsymbol{x}_c(k) \\ \boldsymbol{x}(k) \end{bmatrix},\ \bar{\boldsymbol{u}}(k) = \begin{bmatrix} \boldsymbol{a}(k) \\ 0 \end{bmatrix}$$

$$\bar{A} = \begin{bmatrix} A_1 & 0 \\ 0 & A_1 \end{bmatrix},\ \bar{B} = \begin{bmatrix} B_a & 0 \\ 0 & 0 \end{bmatrix}$$

根据式 (1) 和式 (3) 可以得到

$$\bar{\boldsymbol{x}}(k+1) = \bar{A}\bar{\boldsymbol{x}}(k) + \bar{B}\bar{\boldsymbol{u}}(k) \tag{5}$$

并且可将优化问题 (4) 转化为

$$\min J = \sum_{k=0}^{\infty}\left[\bar{\boldsymbol{x}}^{\mathrm{T}}(k)\bar{Q}\bar{\boldsymbol{x}}(k) + \bar{\boldsymbol{u}}^{\mathrm{T}}(k)\bar{R}\bar{\boldsymbol{u}}(k)\right] \tag{6}$$

其中, $\bar{Q} = \begin{bmatrix} -Q & Q \\ Q & -Q \end{bmatrix} \leq 0$, $\bar{R} = \begin{bmatrix} R & 0 \\ 0 & R \end{bmatrix} \geq 0$.

由式 (6) 可求得最优反馈攻击策略

$$\bar{\boldsymbol{u}}(k) = -(\bar{R} + \bar{B}^{\mathrm{T}}\bar{P}\bar{B})^{-1}\bar{B}^{\mathrm{T}}\bar{P}\bar{A}\bar{\boldsymbol{x}}(k) \tag{7}$$

其中, $\bar{P}$ 通过求解下面的代数 Ricatti 方程得到

$$\bar{P} - \bar{Q} - \bar{A}^{\mathrm{T}}\bar{P}\bar{A} + \bar{A}^{\mathrm{T}}\bar{P}\bar{B}(\bar{R} + \bar{B}^{\mathrm{T}}\bar{P}\bar{B})^{-1}\bar{B}^{\mathrm{T}}\bar{P}\bar{A} = 0 \tag{8}$$

然而, 由于 $\bar{Q} \leq 0$, 优化问题 (6) 不是传统的线性二次型调节器问题, 式 (8) 不一定有唯一正定解, 本文考虑获得式 (8) 的一个特解.

令 $\bar{P} = \begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix}$, P_{11} 是 n 阶方阵, 并将 $\bar{P}$ 代入式 (7) 得

$$\bar{\boldsymbol{u}}(k) = \begin{bmatrix} \boldsymbol{a}(k) \\ 0 \end{bmatrix} = -\begin{bmatrix} \boldsymbol{\psi} \\ 0 \end{bmatrix} \tag{9}$$

其中, $\boldsymbol{\psi} = (R + B_a^{\mathrm{T}}P_{11}B_a)^{-1}B_a^{\mathrm{T}}P_{11}A_1\boldsymbol{x}_c(k) + (R + B_a^{\mathrm{T}}P_{11}B_a)^{-1}B_a^{\mathrm{T}}P_{12}A_1\boldsymbol{x}(k)$, $\boldsymbol{x}(k)$ 是系统在不受攻击情况下的状态.

考虑到攻击策略一般不依赖于受攻击前的系统状态 $\boldsymbol{x}(k)$, 故令 $P_{12} = 0$. 若满足

$$A_1^{\mathrm{T}}P_{11}B_a(R + B_a^{\mathrm{T}}P_{11}B_a)^{-1}B_a^{\mathrm{T}} + P_{11} + Q - A_1^{\mathrm{T}}P_{11}A_1 = 0 \tag{10}$$

$$A_1^{\mathrm{T}}P_{21}B_a(R + B_a^{\mathrm{T}}P_{11}B_a)^{-1}B_a^{\mathrm{T}} + P_{21} - Q - A_1^{\mathrm{T}}P_{21}A_1 = 0 \tag{11}$$

$$P_{22} + Q - A_1^{\mathrm{T}}P_{22}A_1 = 0 \tag{12}$$

那么, $\bar{P} = \begin{bmatrix} P_{11} & 0 \\ P_{21} & P_{22} \end{bmatrix}$ 可以使得式 (8) 成立. 通过求解式 (10)~(12) 获得矩阵 P_{11}, 即可得到最优反馈攻击策略

$$\boldsymbol{a}(k) = -(R + B_a^{\mathrm{T}}P_{11}B_a)^{-1}B_a^{\mathrm{T}}P_{11}A_1\boldsymbol{x}_c(k) \tag{13}$$

结合式 (3) 和式 (13) 可以生成一系列的攻击数据以供训练机器学习算法.

2　攻击信号重构算法

本节利用 IFOA-KELM 算法对攻击信号进行重构.

2.1　核极限学习机

ELM 是一种前馈神经网络[23], 如图 2 所示, 它由输入层、隐藏层和输出层共三层构成.

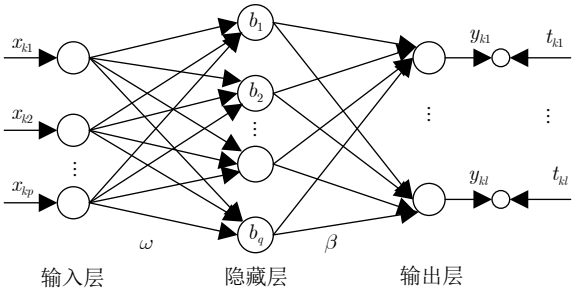


图 2　极限学习机结构图

Fig. 2　The structure of ELM

假设存在 N 个不同的训练样本 $\{(\boldsymbol{x}_k, \boldsymbol{t}_k)\}_{k=1}^{N}$, 其中, $\boldsymbol{x}_k = [x_{k1}, x_{k2}, \cdots, x_{kp}]^{\mathrm{T}} \in \mathbf{R}^p$ 为输入矢量, $\boldsymbol{t}_k = [t_{k1}, t_{k2}, \cdots, t_{kl}]^{\mathrm{T}} \in \mathbf{R}^l$ 为期望输出矢量, $\boldsymbol{y}_k = [y_{k1}, y_{k2}, \cdots, y_{kl}]^{\mathrm{T}} \in \mathbf{R}^l$ 为输出矢量. 将第 i 个隐层神经元与输入层之间的连接权重记作 $\boldsymbol{w}_i = [\omega_{i1}, \omega_{i2}, \cdots, \omega_{ip}]^{\mathrm{T}}$, 并记 $\boldsymbol{w} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \cdots, \boldsymbol{w}_q]^{\mathrm{T}}$. 将第 i 个隐层神经元的阈值记作 b_i. 隐藏层神经元的激活函数为 $\hat{g}(\cdot)$. 将第 i 个隐层神经元与输出层之间的连接权重记作 $\boldsymbol{\beta}_i = [\beta_{i1}, \beta_{i2}, \cdots, \beta_{il}]^{\mathrm{T}}$, 并记 $\boldsymbol{\beta} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\beta}_q]^{\mathrm{T}}$. ELM 的期望目标为 $\sum_{k=1}^{N}\|\boldsymbol{y}_k - \boldsymbol{t}_k\| = 0$, 因此

$$\sum_{i=1}^{q}\boldsymbol{\beta}_i\hat{g}(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{x}_k+b_i)=\boldsymbol{t}_k \tag{14}$$

将上式表示成矩阵形式

$$H\beta=T \tag{15}$$

其中,

$$T=[\boldsymbol{t}_1,\boldsymbol{t}_2,\cdots,\boldsymbol{t}_N]^{\mathrm{T}}$$

$$H=[\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_1),\cdots,\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_N)]^{\mathrm{T}}=$$

$$\begin{bmatrix}\hat{g}(\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}_1+b_1) & \cdots & \hat{g}(\boldsymbol{w}_q^{\mathrm{T}}\boldsymbol{x}_1+b_q)\\ \vdots & \ddots & \vdots\\ \hat{g}(\boldsymbol{w}_1^{\mathrm{T}}\boldsymbol{x}_N+b_1) & \cdots & \hat{g}(\boldsymbol{w}_q^{\mathrm{T}}\boldsymbol{x}_N+b_q)\end{bmatrix}$$

训练 ELM 本质上是为了得到$\boldsymbol{\beta}$, 而根据式 (15) 可得

$$\boldsymbol{\beta}=H^{\dagger}T \tag{16}$$

其中, $H^{\dagger}=H^{\mathrm{T}}\left(HH^{\mathrm{T}}+\dfrac{I_N}{C}\right)^{-1}$, C 为惩罚系数.

在 ELM 训练过程中, $\boldsymbol{w}_i$ 和b_i 是随机给定的, 这可能会导致 ELM 的稳健性和泛化能力变差, 为了解决上述问题, 将隐藏层神经元的随机映射用核映射来代替, 即得到 KELM[24]. 现将 KELM 中采用的核函数定义为

$$\Omega_{\mathrm{ELM}}=HH^{\mathrm{T}}$$

$$\Omega_{\mathrm{ELM}m,n}=\boldsymbol{h}(\boldsymbol{x}_m)\boldsymbol{h}^{\mathrm{T}}(\boldsymbol{x}_n)=\mathcal{K}(\boldsymbol{x}_m,\boldsymbol{x}_n) \tag{17}$$

于是, KELM 的输出 $\boldsymbol{y}$ 可表示为

$$\boldsymbol{y}=\boldsymbol{h}\,(\boldsymbol{x})\boldsymbol{\beta}=\boldsymbol{h}(\boldsymbol{x})\,H^{\mathrm{T}}(HH^{\mathrm{T}}+\frac{I_N}{C})^{-1}T=$$

$$\begin{bmatrix}\mathcal{K}(\boldsymbol{x},\boldsymbol{x}_1)\\ \mathcal{K}(\boldsymbol{x},\boldsymbol{x}_2)\\ \vdots\\ \mathcal{K}(\boldsymbol{x},\boldsymbol{x}_N)\end{bmatrix}^{\mathrm{T}}\left(\Omega_{\mathrm{ELM}}+\frac{I_N}{C}\right)^{-1}T \tag{18}$$

注 1. 根据 Mercer 定理[25], 本文选用高斯核函数作为核函数, 即 $\mathcal{K}(\boldsymbol{x}_a,\boldsymbol{x}_b)=\mathrm{e}^{-\frac{\|\boldsymbol{x}_a-\boldsymbol{x}_b\|^2}{2\sigma^2}}$, 其中$\sigma$ 是带宽.

由式 (18) 可知, KELM 中需要调整的参数只有两个: 核函数参数σ和惩罚因子 C, 它们对于 KELM 的性能起着至关重要的作用. 因此, 如图 3 所示, 本文利用 FOA 择优选取 σ 和C, 并将两个果蝇群体分别称作 σ 群体和C 群体.

2.2 果蝇优化算法

FOA[26] 源于果蝇的觅食行为, 它和粒子群优化算法[27]、鲨鱼优化算法[28]、细菌群体趋药性算法[29] 类似也属于群体智能优化算法的一种. 果蝇具有优于其他物种的嗅觉器官和视觉器官, 它们的觅食原理

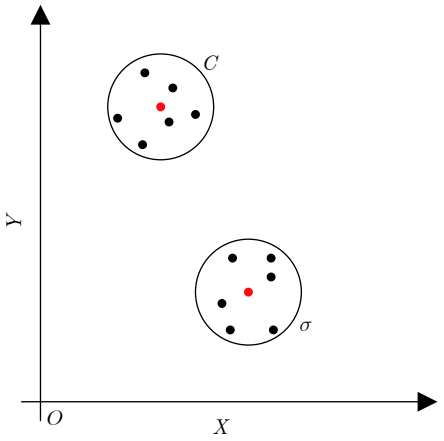


图 3 FOA 优化参数

Fig. 3 The optimization parameter of FOA

如下: 首先借助嗅觉器官搜集空气中弥散的各种气味, 然后发觉目标大致方位并飞往附近区域, 最后利用敏锐的视觉定位目标和群体聚集的具体位置. FOA 具体实现步骤如下:

步骤 1. 初始化果蝇群体大小, FOA 最大迭代次数以及果蝇群体位置.

步骤 2. 每个果蝇个体按照随机给定的方向和范围搜寻目标位置.

步骤 3. 由于果蝇个体事先不知道目标所在的位置, 因此以原点作为参考点, 计算个体到原点的距离, 并将该距离的倒数作为个体味道浓度判定值.

步骤 4. 将味道浓度判定值代入适应度函数求得个体的适应度值.

步骤 5. 确定具有最优适应度值的个体所在位置, 以供其他个体借助视觉飞向此位置.

步骤 6. 重复执行步骤 2 ~ 5, 直至迭代次数超限, 结束算法.

上述基本的 FOA 容易陷入局部最优解, 对此本文对其进行改进. 理论上, 可以通过将每次迭代寻优过程中果蝇群体的初始位置随机 "小范围"地置于另一新位置以及增大果蝇个体寻优范围的方式帮助寻优过程有效地跳出局部死循环. 值得注意的是, 这里的 "小范围"需要保证经上一次迭代得到的最优个体位置在本次迭代个体寻优范围内, 进而使本次迭代能够得到比上次更优的结果. 最终, 得到 IFOA.

2.3 基于 IFOA-KELM 的攻击信号重构算法

利用 IFOA 先对 KELM 中核函数参数 σ 和惩罚因子C两个参数的初始值进行优化, 再进行 KELM 的训练. 具体的 IFOA-KELM 算法步骤如下:

步骤 1. 初始化果蝇群体大小 *particlesize*,

FOA 最大迭代次数 Max_num, 两个果蝇群体位置 $\boldsymbol{X\_axis}=[X\_axis_1, X\_axis_2]$ 和 $\boldsymbol{Y\_axis}=[Y\_axis_1, Y\_axis_2]$ 以及最优适应度值 $Smellbest$.

步骤 2. 设定果蝇个体初始寻优半径为 (r_1, r_2), 并随机给定果蝇个体搜寻方向 $(2\times Random-1)\in[-1,1)$ 以确定个体接下来飞向的位置

$$\begin{cases} X_{i1}=X\_axis_1+r_1\times(2\times Random-1)\\ Y_{i1}=Y\_axis_1+r_1\times(2\times Random-1)\\ X_{i2}=X\_axis_2+r_2\times(2\times Random-1)\\ Y_{i2}=Y\_axis_2+r_2\times(2\times Random-1)\end{cases}\tag{19}$$

其中, i 表示第 i 组果蝇个体, 由 C 群体中的一个个体和 σ 群体中的一个个体组成. (X_{i1}, Y_{i1}) 表示 C 群体中第 i 个个体的位置, (X_{i2}, Y_{i2}) 表示 σ 群体中第 i 个个体的位置, $Random$ 是一个 $[0,1)$ 范围服从均匀分布的随机数, $i=1,2,\cdots,particlesize$.

步骤 3. 计算第 i 组果蝇个体分别与 $(0,0)$ 之间的距离 $\boldsymbol{D}_i=[D_{i1}, D_{i2}]$, 其中,

$$\begin{cases} D_{i1}=\sqrt{X_{i1}^2+Y_{i1}^2}\\ D_{i2}=\sqrt{X_{i2}^2+Y_{i2}^2}\end{cases}$$

以及相应的味道浓度判定值 $\boldsymbol{S}_i=[S_{i1}, S_{i2}]$, 其中,

$$\begin{cases} S_{i1}=\dfrac{1}{D_{i1}}\\ S_{i2}=\dfrac{1}{D_{i2}}\end{cases}$$

步骤 4. 将味道浓度判定值 $\boldsymbol{S}_i$ 代入如下适应度函数以得到适应度值 $Smell_i$

$$Smell_i=\frac{1}{2N}\sum_{k=1}^{N}\sum_{r=1}^{l}(y_r^i-t_r^i)\tag{20}$$

其中, N 为训练样本数.

步骤 5. 确定具有最优适应度值的果蝇个体组

$$[bestSmell, bestIndex]=\min(\boldsymbol{Smell})\tag{21}$$

其中, $\boldsymbol{Smell}=[Smell_1,\cdots,Smell_{particlesize}]$, $bestSmell$ 表示最优适应度值, $bestIndex$ 表示得到最优适应度值的果蝇个体组号.

步骤 6. 判断 $bestSmell<Smellbest$ 是否成立, 若成立, 按照式 (22) 更新最优适应度值 $Smellbest$ 及与之对应的最优个体位置坐标 ($\boldsymbol{Xbest}$, $\boldsymbol{Ybest}$), 并将果蝇个体寻优半径置为初始值 r, 此外, 将最优味道浓度值 $\boldsymbol{S}_{\text{bestIndex}}$ 记录至数组 $\boldsymbol{SS}$

$$\begin{cases} Smellbest=bestSmell\\ \boldsymbol{X\_axis}=\boldsymbol{Xbest}=\boldsymbol{X}_{\text{bestIndex}}\\ \boldsymbol{Y\_axis}=\boldsymbol{Ybest}=\boldsymbol{Y}_{\text{bestIndex}}\\ \boldsymbol{SS}=\boldsymbol{S}_{\text{bestIndex}}\end{cases}\tag{22}$$

其中, $\boldsymbol{X}_i=[X_{i1}, X_{i2}]$, $\boldsymbol{Y}_i=[Y_{i1}, Y_{i2}]$.

否则, 按照式 (23) 更新下次迭代的初始果蝇群体位置 $\boldsymbol{X\_axis}$, $\boldsymbol{Y\_axis}$ 和果蝇个体寻优半径 r

$$\begin{cases} \boldsymbol{X\_axis}=\boldsymbol{Xbest}+\dfrac{\sqrt{2}}{2}\times(2\times Random-1)\times\boldsymbol{r}\\ \boldsymbol{Y\_axis}=\boldsymbol{Ybest}+\dfrac{\sqrt{2}}{2}\times(2\times Random-1)\times\boldsymbol{r}\\ \boldsymbol{r}\leftarrow 1.5\times\boldsymbol{r}\end{cases}\tag{23}$$

其中, $\boldsymbol{r}=[r_1, r_2]$.

将本次迭代产生的最优个体位置作为下次迭代的初始果蝇群体位置, 即满足 $\boldsymbol{X\_axis}=\boldsymbol{Xbest}$ 和 $\boldsymbol{Y\_axis}=\boldsymbol{Ybest}$.

步骤 7. 重复步骤 2 ~ 6, 为了防止 KELM过拟合, 需要满足 $Smellbest$ 不超过给定值 θ; 此外, 当前迭代次数不得超过 Max_num. 否则, 跳出循环, 并将 SS 的最后记录用于训练 KELM.

步骤 8. 利用训练好的 KELM 对需要测试的样本数据进行预测, 算法结束.

注 2. 由式 (20) 可知, 适应度值实为 KELM 算法的均方误差损失项, 如果一味地追求训练集上误差损失最小化, 则会导致所训练的算法过分拟合训练集, 从而导致所训练算法在测试集上表现变差. 所以, 需要针对 $Smellbest$ 给出合适的下界 θ 以防止 IFOA-KELM 算法过拟合. 这里合适的 θ 值需要通过反复试验来获取.

以单个果蝇群体为例, 具体的 IFOA 寻优过程如图 4 所示. 其中, "×"代表局部最优解, 阴影区域表示最优位置坐标变化范围. 由图 4 可看出, 当寻优过程陷入局部最优解时, 会通过移动最优位置坐标以及放大搜索半径的方式设法跳出局部最优解.

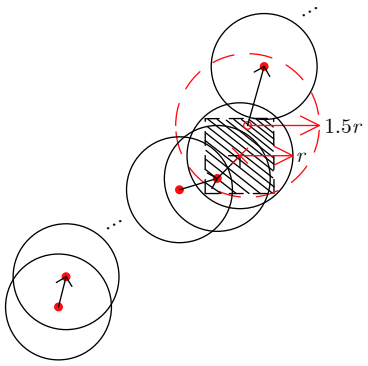


图 4　IFOA 寻优过程

Fig. 4　The optimization process of IFOA

基于上述 IFOA-KELM 算法得到重构的攻击信号 $g(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))$.

注 3. 径向基函数 $g(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))$ 满足 Lipschitz 条件.

将 $g(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))$ 与真实攻击信号 $a(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))$ 之间的偏差记为 $\boldsymbol{\omega}(k)$, 由前馈神经网络的万能逼近特性[30] 得

$$\|\boldsymbol{\omega}(k)\| = \|a(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k)) - g(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))\| \le \gamma \tag{24}$$

其中, $\gamma > 0$ 为偏差上界.

为保证受攻击系统的安全运行, 将系统 (2) 的状态反馈控制策略更新为 $\boldsymbol{u}(k)$, 此时系统的状态方程可表示为

$$\boldsymbol{x}_c(k+1) = A\boldsymbol{x}_c(k) + B\boldsymbol{u}(k) + a(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k)) = f^*(\boldsymbol{x}_c(k), \boldsymbol{u}(k)) \tag{25}$$

利用 $g(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))$ 替代式 (25) 中的真实攻击信号 $a(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))$ 得到标称模型

$$\boldsymbol{x}_c(k+1) = A\boldsymbol{x}_c(k) + B\boldsymbol{u}(k) + g(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k)) = f(\boldsymbol{x}_c(k), \boldsymbol{u}(k)) \tag{26}$$

显然, $f(\boldsymbol{x}, \boldsymbol{u})$ 满足 Lipschitz 条件, 并设 $f(\boldsymbol{x}, \boldsymbol{u})$ 关于 $\boldsymbol{x}$ 的 Lipschitz 常数为 $L_f \in (0, \infty)$.

由式 (24) 可知, $f^*(\boldsymbol{x}_c(k), \boldsymbol{u}(k))$ 与 $f(\boldsymbol{x}_c(k), \boldsymbol{u}(k))$ 满足

$$\|f^*(\boldsymbol{x}_c(k), \boldsymbol{u}(k)) - f(\boldsymbol{x}_c(k), \boldsymbol{u}(k))\| = \|\boldsymbol{\omega}(k)\| \le \gamma \tag{27}$$

因此, 可将式 (25) 看作是包含有界外部扰动的不确定系统. 由式 (24) 可知, 所得受攻击系统的标称模型与真实受攻击系统模型之间存在一定的误差, 此时如果继续使用状态反馈控制策略, 仅仅改变控制律的状态反馈增益矩阵, 不能够很好地应对该误差进而保证被控系统的稳定性. 而 MPC 具有良好的内在鲁棒性, 并且可以结合合适的收缩约束条件很好地处理这一问题. 因此, 在得到式 (26) 之后, 假设攻击者不再更新其攻击策略, 使用 MPC 策略对受攻击系统进行控制以保证系统安全运行.

3 模型预测控制器

模型预测控制是一种基于模型的开环最优控制策略[31−32], 通过在线求解有限时域优化控制问题计算预测状态和未来的控制输入.

为了获取使得受攻击系统安全运行的控制信号 $\boldsymbol{u}(k)$, 本文求解如下有限时域 MPC 优化问题

$$\begin{aligned} \min_{\boldsymbol{u}_M(k)} \quad & J_{H_p}(\boldsymbol{x}_c(k), \boldsymbol{U}_M(k)) \\ \text{s.t.} \quad & \boldsymbol{x}_c(k+j+1|k) = f(\boldsymbol{x}_c(k+j|k), \boldsymbol{u}_M(k+j|k)) \\ & \qquad j \in [0, H_p - 1] \\ & \boldsymbol{x}_c(k+H_p|k) \in \Omega \end{aligned} \tag{28}$$

其中, 决策变量 $\boldsymbol{u}_M(k+j|k)$ 定义了模型预测控制器在 k 时刻预测的 $k+j$ 时刻控制输入, 因此 $\boldsymbol{U}_M(k) = \{\boldsymbol{u}_M(k|k), \cdots, \boldsymbol{u}_M(k+H_p-1|k)\}$ 表示系统在未来预测时域 H_p 内的控制输入序列; $\boldsymbol{x}_c(k+j|k)$ 表示在 $\boldsymbol{U}_M(k)$ 作用下标称系统 $k+j$ 时刻的预测状态; Ω 表示终端状态约束集. 将优化问题 (28) 的最优解表示为 $\hat{\boldsymbol{U}}_M(k) = \{\hat{\boldsymbol{u}}_M(k|k), \cdots, \hat{\boldsymbol{u}}_M(k+H_p-1|k)\}$, 相应的预测状态为 $\hat{\boldsymbol{X}}_M(k) = \{\hat{\boldsymbol{x}}_c(k|k), \cdots, \hat{\boldsymbol{x}}_c(k+H_p|k)\}$, 其中 $\hat{\boldsymbol{x}}_c(k|k) = \boldsymbol{x}_c(k) = \boldsymbol{x}_c(k|k)$, 与此相对应的最优成本记为 $\hat{J}_{H_p}(k)$. 系统在 k 时刻的实际输入为 $\boldsymbol{u}_M(k) = \hat{\boldsymbol{u}}_M(k|k)$. 现将成本函数 $J_{H_p}(\boldsymbol{x}_c(k), \boldsymbol{U}_M(k))$ 定义为

$$J_{H_p}(\boldsymbol{x}_c(k), \boldsymbol{U}_M(k)) = \sum_{j=0}^{H_p-1} L(\boldsymbol{x}_c(k+j|k), \boldsymbol{u}_M(k+j|k)) + V(\boldsymbol{x}_c(k+H_p|k)) \tag{29}$$

其中, $L(\boldsymbol{x}, \boldsymbol{u})$ 为阶段成本函数, $V(\boldsymbol{x}_c(k+H_p|k)$ 为终端成本函数.

引理 1[33]**.** 基于标称模型的最优预测状态 $\hat{\boldsymbol{x}}_c(k+j|k)$ $(j \ge 1)$ 与真实状态 $\boldsymbol{x}_c(k+j)$ $(j \ge 1)$ 之间的偏差满足以下关系

$$\|\hat{\boldsymbol{x}}_c(k+j|k) - \boldsymbol{x}_c(k+j)\| \le \frac{L_f^j - 1}{L_f - 1}\gamma \tag{30}$$

针对阶段成本函数及终端集, 本文给出如下假设:

假设 1. 假设阶段成本函数 $L(\boldsymbol{x}, \boldsymbol{u})$ 满足 $L(0, 0) = 0$ 并且是 Lipschitz 连续的, 记 $L(\boldsymbol{x}, \boldsymbol{u})$ 相对 $\boldsymbol{x}$ 的 Lipschitz 常数为 $L_c \in (0, \infty)$; 另外, 存在常数 $\varphi > 0$ 和 $\sigma \ge 1$ 使得 $L(\boldsymbol{x}, \boldsymbol{u}) \ge \varphi\|(\boldsymbol{x}, \boldsymbol{u})\|^\sigma$ 成立, 即

$$|L(\boldsymbol{x}_1, \boldsymbol{u}) - L(\boldsymbol{x}_2, \boldsymbol{u})| \le L_c\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \tag{31}$$

假设 2. 定义 Φ 是对于标称系统 (26) 的一个正不变集, 且 $\Omega \subseteq \Phi$. 存在局部控制器 $\boldsymbol{u}_M(k) = h(\boldsymbol{x}_c(k))$ 以及相关 Lyapunov 函数使得

1) 对于 $\boldsymbol{x}_c(k) \in \Phi$, $V(f(\boldsymbol{x}_c(k), h(\boldsymbol{x}_c(k)))) - V(\boldsymbol{x}_c(k)) \le -L(\boldsymbol{x}_c(k), h(\boldsymbol{x}_c(k)))$.

2) 终端成本函数 $V(\boldsymbol{x}, \boldsymbol{u})$ 在 Φ 内是 Lipschitz 连续的, 相对 $\boldsymbol{x}$ 的 Lipschitz 常数记为 L_v, 即

$$|V(\boldsymbol{x}_1, \boldsymbol{u}) - V(\boldsymbol{x}_2, \boldsymbol{u})| \le L_v\|\boldsymbol{x}_1 - \boldsymbol{x}_2\| \quad \forall \boldsymbol{x}_1, \boldsymbol{x}_2 \in \Phi$$

其中, $\Phi = \{\boldsymbol{x} \in \mathbf{R}^n : V(\boldsymbol{x}) \le \alpha\}$.

假设 3. 集合 $\Omega = \{\boldsymbol{x} \in \mathbf{R}^n : V(\boldsymbol{x}) \le \alpha_v\}$ 满足 $\forall \boldsymbol{x} \in \Phi$, $f(\boldsymbol{x}, h(\boldsymbol{x})) \in \Omega$.

引理 2[33]**.** 假设 k 时刻优化问题 (28) 存在最优解 $\hat{\boldsymbol{U}}_M(k) = \{\hat{\boldsymbol{u}}_M(k|k), \hat{\boldsymbol{u}}_M(k+1|k), \cdots, \hat{\boldsymbol{u}}_M(k+H_p-1|k)\}$, 据此构造 $k+1$ 时刻的解 $\tilde{\boldsymbol{U}}_M(k+1) = \{\hat{\boldsymbol{u}}_M(k+1|k), \cdots, \hat{\boldsymbol{u}}_M(k+H_p-1|k), h(\tilde{\boldsymbol{x}}(k+H_p|k+1))\}$, 即

$$\tilde{\boldsymbol{u}}_M(k+j+1)=\begin{cases}\hat{\boldsymbol{u}}_M(k+j+1|k), & j\in[0,H_p-2]\\ h(\tilde{\boldsymbol{x}}(k+j+1|k+1)), & j=H_p-1\end{cases}$$

则由标称模型 (26) 得到的 $k+1$ 时刻预测状态序列为 $\tilde{\boldsymbol{X}}_M(k+1)=\{\tilde{\boldsymbol{x}}_c(k+1|k+1),\tilde{\boldsymbol{x}}_c(k+2|k+1),\cdots,\tilde{\boldsymbol{x}}_c(k+H_p+1|k+1)\}$, 其中 $\tilde{\boldsymbol{x}}_c(k+1|k+1)=\boldsymbol{x}_c(k+1|k+1)$. 此时, 预测状态 $\tilde{\boldsymbol{x}}_c(k+H_p|k+1)$ 和 $\hat{\boldsymbol{x}}_c(k+H_p|k)$ 的偏差满足

$$\|\tilde{\boldsymbol{x}}_c(k+H_p|k+1)-\hat{\boldsymbol{x}}_c(k+H_p|k)\|\le L_f^{H_p-1}\gamma \quad (32)$$

定理 1. 当闭环系统 (25) 的参数满足假设 1 ~ 3 以及 $\gamma\le\dfrac{\alpha-\alpha_v}{L_vL_f^{N-1}}$ 时, 优化问题 (28) 是迭代可行的, 并且闭环系统 (25) 是 ISS 的.

证明.

1) 可行性. 由假设 2 和引理 2 得

$$V(\tilde{\boldsymbol{x}}_c(k+H_p|k+1))\le V(\hat{\boldsymbol{x}}_c(k+H_p|k))+L_vL_f^{H_p-1}\gamma\le \alpha_v+L_vL_f^{H_p-1}\gamma\le\alpha$$

因此, $\tilde{\boldsymbol{x}}_c(k+H_p|k+1)\in\Phi$. 再由假设 3 可知, 存在局部控制器 $h(\tilde{\boldsymbol{x}}_c(k+H_p|k+1))$ 使得 $\tilde{\boldsymbol{x}}_c(k+H_p+1|k+1)\in\Omega$. 于是, 优化问题 (28) 是迭代可行的.

2) 稳定性. 由假设 1 ~ 3 以及引理 1 和引理 2 得

$$\begin{aligned}&\hat{J}_{H_p}(k+1)-\hat{J}_{H_p}(k)\le\\&\sum_{i=0}^{H_p-1}\{L(\tilde{\boldsymbol{x}}(k+i+1|k+1),\tilde{\boldsymbol{u}}(k+i+1|k+1))-\\&L(\hat{\boldsymbol{x}}(k+i|k),\hat{\boldsymbol{u}}(k+i|k))\}+\\&V(\tilde{\boldsymbol{x}}(k+H_p+1|k+1))-V(\hat{\boldsymbol{x}}(k+H_p|k))=\\&\sum_{i=0}^{H_p-2}\{L(\tilde{\boldsymbol{x}}(k+i+1|k+1),\tilde{\boldsymbol{u}}(k+i+1|k+1))-\\&L(\hat{\boldsymbol{x}}(k+i+1|k),\hat{\boldsymbol{u}}(k+i+1|k))\}+\\&L(\tilde{\boldsymbol{x}}(k+H_p|k+1),h(\tilde{\boldsymbol{x}}(k+H_p|k)))-\\&L(\boldsymbol{x}(k|k),\boldsymbol{u}(k|k))+V(\tilde{\boldsymbol{x}}(k+H_p+1|k+1))-\\&V(\hat{\boldsymbol{x}}(k+H_p|k))\le\\&L_c\frac{1-L_f^{H_p-1}}{1-L_f}\gamma+L(\tilde{\boldsymbol{x}}(k+H_p|k+1),h(\tilde{\boldsymbol{x}}(k+H_p|k)))+\\&V(\tilde{\boldsymbol{x}}(k+H_p+1|k+1))-V(\tilde{\boldsymbol{x}}(k+H_p|k+1))+\\&L_vL_f^{H_p-1}\gamma-L(\boldsymbol{x}(k|k),\boldsymbol{u}(k|k))\le\\&L_c\frac{1-L_f^{H_p-1}}{1-L_f}\gamma+L_vL_f^{H_p-1}\gamma-L(\boldsymbol{x}(k|k),\boldsymbol{u}(k|k))=\\&L_Z\times\gamma-\varphi\|\boldsymbol{x}(k)\|^{\sigma}\end{aligned}$$

其中, $L_Z=L_c\dfrac{1-L_f^{H_p-1}}{1-L_f}+L_vL_f^{H_p-1}$. 因此, 闭环系统 (25) 是 ISS 的. □

4　数值仿真及结果分析

本节将通过数值算例验证本文提出的基于机器学习的安全控制策略的有效性. 考虑如图 5 所示的弹簧–质量–阻尼系统. 其中, m 表示物体的质量, u 表示作用在物体上的力, s 表示物体的位移, v 表示物体的运动速度, K_l 表示弹簧的弹性系数, K_d 表示阻尼器的阻尼系数.

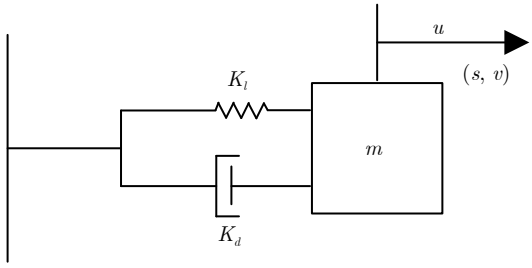


图 5　弹簧–质量–阻尼系统结构图

Fig. 5　The structure of spring-quality-damping system

未受攻击的系统模型为

$$\begin{cases}\boldsymbol{x}(k+1)=\begin{bmatrix}0.9819 & 0.1716\\ -0.1716 & 0.7245\end{bmatrix}\boldsymbol{x}(k)+\begin{bmatrix}0.0181\\ 0.1716\end{bmatrix}u(k)\\ u(k)=\begin{bmatrix}-0.1613 & -1.1854\end{bmatrix}\boldsymbol{x}(k)\end{cases} \quad (33)$$

其中, $\boldsymbol{x}(k)=[s^{\mathrm{T}}(k),v^{\mathrm{T}}(k)]^{\mathrm{T}}$.

假设攻击者按照设计的最优攻击策略 (3) 和 (13) 对系统 (33) 进行攻击, 其中, $B_a=I_2$. 令 $Q=\begin{bmatrix}1.2786 & -0.4926\\ -0.4926 & 3.2637\end{bmatrix}$, $R=1$, 由式 (10) ~ (12) 得到

$$P_{11}=\begin{bmatrix}-20.1258 & -5.9459\\ -5.9459 & -8.3562\end{bmatrix}$$

将 P_{11} 代入式 (13) 得到受攻击后的系统模型

$$\begin{cases}\boldsymbol{x}_c(k+1)=\begin{bmatrix}0.9790 & 0.1502\\ -0.1993 & 0.5210\end{bmatrix}\boldsymbol{x}_c(k)+\boldsymbol{a}(k)\\ \boldsymbol{a}(k)=\begin{bmatrix}0.0267 & 0.0257\\ 0.2532 & 0.2443\end{bmatrix}\boldsymbol{x}_c(k)\end{cases} \quad (34)$$

设系统的初始状态为 $\boldsymbol{x}_0=[0.8,0]^{\mathrm{T}}$, 则系统在受到攻击前后的状态变化曲线如图 6 所示. 从图中可以看出, 受攻击后的系统状态发散.

通过随机选取 16 组不同的初始状态生成如图 7 所示的 1 600 组数据 $(\boldsymbol{x}_c(k),\boldsymbol{a}(k))$. 将 $\boldsymbol{x}_c(k)$ 作为输入样本, $\boldsymbol{a}(k)$ 作为期望输出进行 IFOA-KELM 的训练, 经 IFOA 优化得到的 KELM 初始化参数为 $C_{\text{best}}=10\,469$, $\sigma_{\text{best}}=8.5945$.

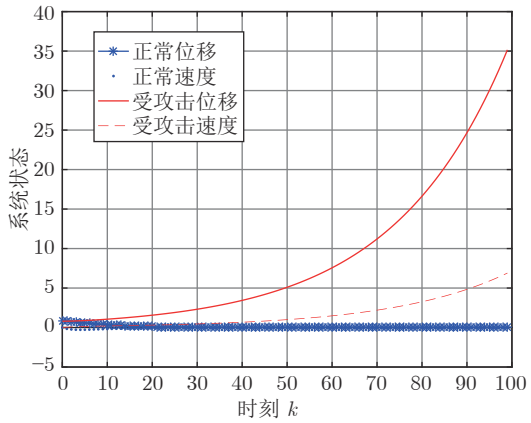


图 6 系统状态轨迹

Fig. 6 The state of the system

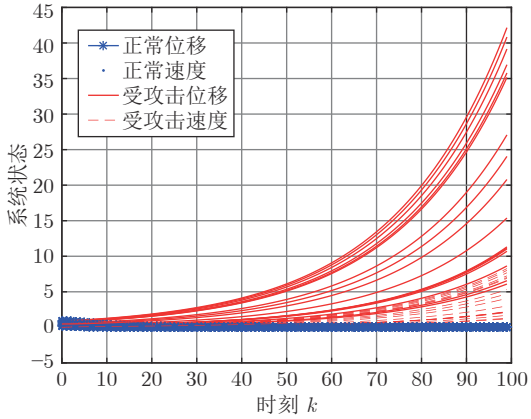


图 7 训练样本

Fig. 7 The training sample

基于最优攻击信号 $\boldsymbol{a}(k)=\begin{bmatrix}0.0267 & 0.0257\\ 0.2532 & 0.2443\end{bmatrix}\boldsymbol{x}_c(k)$ 随机生成 400 组数据分别对所训练的 IFOA-KELM、FOA-KELM、PSO-BP 和 LS-SVM 进行测试, 得到重构的攻击信号与真实攻击信号之间的绝对误差, 如图 8 ~ 11 所示.

对比发现, IFOA-KELM 的学习效果要优于 FOA-KELM、PSO-BP 和 LS-SVM. 因此, 本文选用 IFOA-KELM 对攻击信号进行重构, 且 $\gamma=10^{-4}$.

为了对比 IFOA-KELM 与 FOA-KELM 的初始参数值优化性能, 将 IFOA 与 FOA 同样迭代 50 次后得到二者的最优适应度值变化曲线, 如图 12 所示.

由图 12 可以看出, FOA-KELM 的初始参数优化过程容易陷入局部最优解, 收敛速度更慢; 而 IFOA-KELM 的初始参数优化过程能够及时跳出局部最优解, 继续寻找最优解, 收敛速度更快. 此

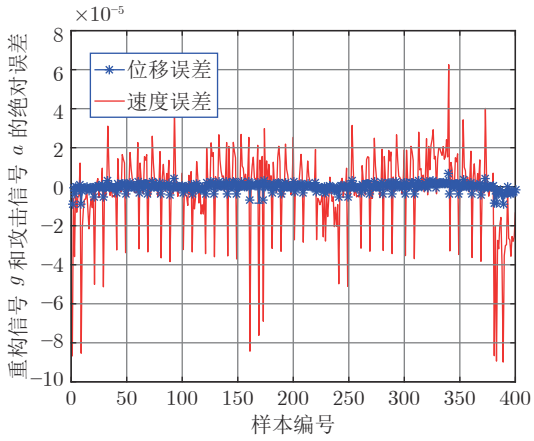


图 8 IFOA-KELM 测试样本绝对误差

Fig. 8 The error between the real attack and the attack learned by IFOA-KELM

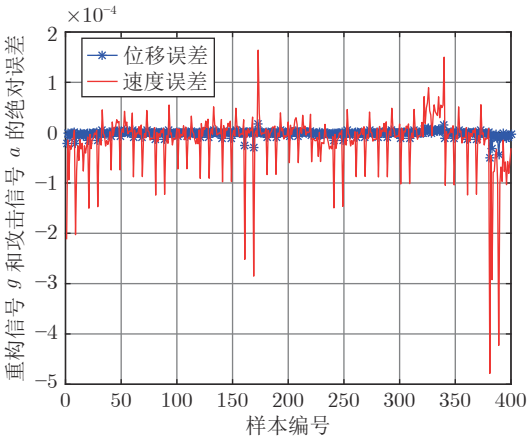


图 9 FOA-KELM 测试样本绝对误差

Fig. 9 The error between the real attack and the attack learned by FOA-KELM

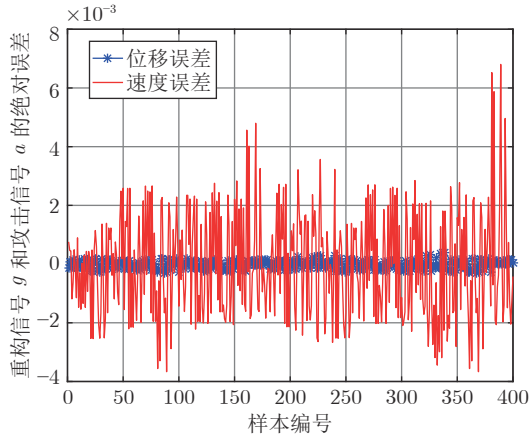


图 10 PSO-BP 测试样本绝对误差

Fig. 10 The error between the real attack and the attack learned by PSO-BP

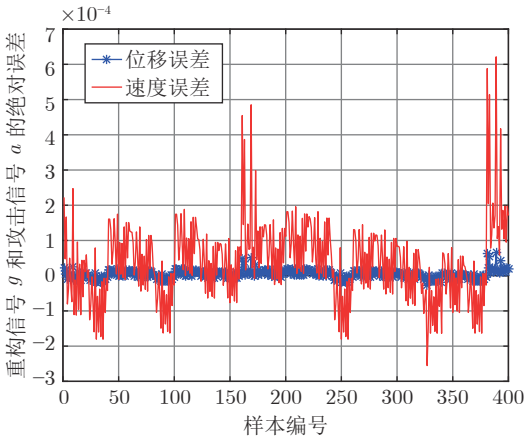


图 11　LSSVM 测试样本绝对误差

Fig. 11　The error between the real attack and the attack learned by LSSVM

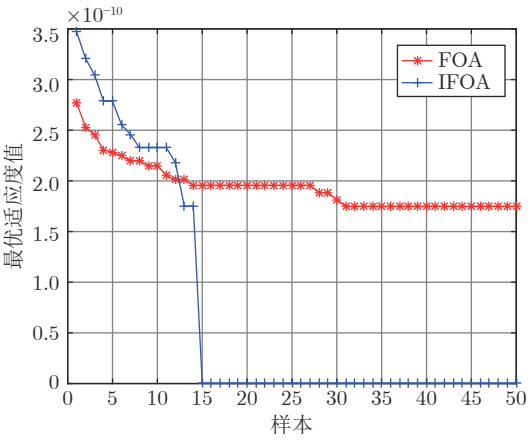


图 12　IFOA 和 FOA 最优适应度值变化曲线

Fig. 12　The Smellbest of IFOA and FOA

外, 经过实验发现, 当 $Smellbest < 10^{-11}$ 时, 利用 IFOA 得到的最优参数 C 和 σ 训练的 KELM 会产生过拟合, 因此 IFOA-KELM 算法中的 θ 取 10^{-11}.

为了保证受攻击系统的安全性, 将系统的控制方式改为 MPC. 系统标称模型 (26) 的 Lipschitz 常数为 $L_f = 1.0539$. 根据式 (29), 将 MPC 的阶段成本函数取为 $L(\boldsymbol{x}, \boldsymbol{u}) = \|\boldsymbol{x}\|_{\bar{Q}} + \|\boldsymbol{u}\|_{\bar{R}}$, 其中 $\bar{Q} = 0.5I_2$, $\bar{R} = 0.1$, 则阶段成本函数 $L(\boldsymbol{x}, \boldsymbol{u})$ 的 Lipschitz 常数为 $L_c = 0.5$; 将终端成本函数取为 $V(\boldsymbol{x}) = \sqrt{\boldsymbol{x}^{\mathrm{T}} P \boldsymbol{x}}$, 其中, $P = \begin{bmatrix} 4.6958 & 1.4098 \\ 1.4098 & 1.3978 \end{bmatrix} > 0$, 并由此得到 $L_v = 2.2839$. 选取 $\alpha = 0.3$, $\alpha_v = 0.2$, 分别对应正不变集 $\Phi = \{\boldsymbol{x} \in \mathbf{R}^n : V(\boldsymbol{x}) \leq 0.3\}$, 终端状态约束集 $\Omega = \{\boldsymbol{x} \in \mathbf{R}^n : V(\boldsymbol{x}) \leq 0.2\}$. 当系统运行进入到终端域 Ω 时, 采用局部状态反馈控制器 $u_M(k) = h(\boldsymbol{x}_c(k)) = [-2.9281 \ -1.8531]\boldsymbol{x}_c(k)$. 此时, 条件 $\gamma \leq \dfrac{\alpha - \alpha_v}{L_v L_f^{N-1}} = 0.03$ 成立. 因此, 根据定理 1 可知, 闭环系统是 ISS 的. 对受攻击后的系统分别使用原有控制策略和 MPC 策略进行控制, 系统的状态变化曲线如图 13 所示.

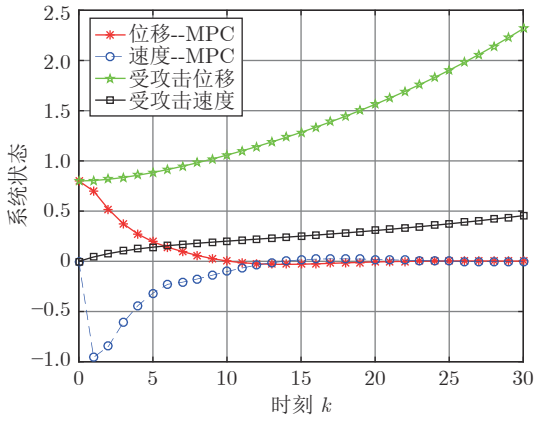


图 13　受攻击系统引入 MPC 前后的状态轨迹

Fig. 13　The state trajectory of the attacked system with MPC and without MPC

在 MPC 策略中, 系统受到的真实攻击信号 $a(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))$ 与经 IFOA-KELM 重构的攻击信号 $g(\boldsymbol{x}_c(k), K\boldsymbol{x}_c(k))$ 之间的绝对误差如图 14 所示.

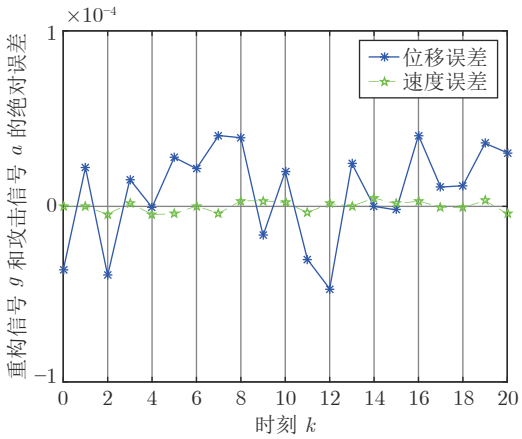


图 14　真实攻击信号与重构攻击信号之间的误差

Fig. 14　The error between the real attack and the learned

5　结论

本文针对受攻击的信息物理系统设计了一种基于机器学习的安全控制方法. 首先, 提出了一种 IFOA-KELM 算法对攻击信号进行重构. 然后, 对受攻击系统设计 MPC 策略, 并给出了使被控系统输入到状态稳定的条件. 此外, 从攻击者角度建立

了优化模型, 得到最优攻击策略以生成足够的受攻击数据. 最后, 利用以弹簧−质量−阻尼系统作为物理对象的 CPS 进行数值仿真, 将 IFOA-KELM、FOA-KELM、LS-SVM 和 PSO-BP 的攻击信号重构效果进行对比. 仿真结果表明 IFOA-KELM 的初始参数优化阶段能够有效解决 FOA-KELM 初始参数优化阶段容易陷入局部最优的问题, 加快整个寻优过程的收敛速度, 并且 IFOA-KELM 相较其他三种算法能够获得更好的拟合效果; 此外, 还验证了本文所提安全控制策略的有效性.

另外, 本文所提的基于机器学习的攻击信号重构算法和 MPC 算法均需要较大的计算资源. 因此, 本文接下来将考虑利用云计算实现上述算法, 并进一步考虑云控制系统[34] 的安全问题.

References

1 Lee J, Bagheri B, Kao H A. A cyber-physical systems architecture for industry 4.0-based manufacturing systems. *Manufacturing Letters*, 2015, **3**: 18−23

2 Li Hong-Yang, Wei Mu-Heng, Huang Jie, Qiu Bo-Hua, Zhao Ye, Luo Wen-Cheng, He Xiao, He Xiao. Survey on cyber-physical systems. *Acta Automatica Sinica*, 2019, **45**(1): 37−50
(李洪阳, 魏慕恒, 黄洁, 邱伯华, 赵晔, 骆文城, 何晓, 何潇. 信息物理系统技术综述. 自动化学报, 2019, **45**(1): 37−50)

3 Inoue J, Yamagata Y, Chen Y, Poskitt C, Sun J. Anomaly detection for a water treatment system using unsupervised machine learning. In: Proceedings of the 2017 IEEE International Conference on Data Mining Workshops. New Orleans, LA, USA: IEEE, 2017. 1058−1065

4 Li D, Chen D C, Goh J, Ng S K. Anomaly detection with generative adversarial networks for multivariate time series. arXiv: 1809.04758, 2018.

5 He H B, Yan J. Cyber-physical attacks and defences in the smart grid: A survey. *IET Cyber-Physical Systems: Theory and Applications*, 2016, **1**(1): 13−27

6 Xia Yuan-Qing, Yan Ce, Wang Xiao-Jing, Song Xiang-Hui. Intelligent transportation cyber-physical cloud control systems. *Acta Automatica Sinica*, 2019, **45**(1): 132−142
(夏元清, 闫策, 王笑京, 宋向辉. 智能交通信息物理融合云控制系统. 自动化学报, 2019, **45**(1): 132−142)

7 Wang H J, Zhao H T, Zhang J, Ma D T. Survey on unmanned aerial vehicle networks: A cyber physical system perspective. arXiv: 1812.06821, 2018.

8 Liu Ting, Tian Jue, Wang Jia-Zhou, Wu Hong-Yu, Sun Li-Min, Zhou Ya-Dong, Shen Chao, Guan Xiao-Hong. Integrated security threats and defense of cyber-physical systems. *Acta Automatica Sinica*, 2019, **45**(1): 5−24
(刘烃, 田决, 王稼舟, 吴宏宇, 孙利民, 周亚东, 沈超, 管晓宏. 信息物理融合系统综合安全威胁与防御研究. 自动化学报, 2019, **45**(1): 5−24)

9 Wolf M, Serpanos D. Safety and security in cyber-physical systems and internet-of-things systems. *Proceedings of the IEEE*, 2018, **106**(1): 9−20

10 De Persis C, Tesi P. Input-to-state stabilizing control under denial-of-service. *IEEE Transactions on Automatic Control*, 2015, **60**(11): 2930−2944

11 Liu K, Guo H, Zhang Q R, Xia Y Q. Distributed secure filtering for discrete-time systems under Round-Robin protocol and deception attacks. *IEEE Transactions on Cybernetics*, 2020, **50**(8): 3571−3580

12 Peng L H, Shi L, Cao X, Sun C Y. Optimal attack energy allocation against remote state estimation. *IEEE Transactions on Automatic Control*, 2018, **63**(7): 2199−2205

13 Zhang Q R, Liu K, Xia Y Q, Ma A Y. Optimal stealthy deception attack against cyber-physical systems. *IEEE Transactions on Cybernetics*, 2020, **50**(9): 3963−3972

14 Zhu Q Y, Basar T. Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: Games-in-games principle for optimal cross-layer resilient control systems. *IEEE Control Systems Magazine*, 2015, **35**(1): 46−65

15 Vu N H, Choi Y S, Choi M. DDoS attack detection using K-nearest neighbor classifier method. In: Proceedings of the 4th IASTED International Conference on Telehealth/Assistive Technologies. Baltimore, Maryland, USA, 2008. 248−253

16 Kumar P G, Devaraj D. Intrusion detection using artificial neural network with reduced input features. *ICTACT Journal on Soft Computing*, 2010: 30−36

17 Nawaz R, Shahid M A, Qureshi I M, Mehmood M H. Machine learning based false data injection in smart grid. In: Proceedings of the 1st International Conference on Power, Energy and Smart Grid. Mirpur, Azad Kashmir, Pakistan, 2018. 1−6

18 Esmalifalak M, Liu L, Nguyen N, Zheng R. Detecting stealthy false data injection using machine learning in smart grid. *IEEE Systems Journal*, 2017, **11**(3): 1644−1652

19 Kiss I, Genge B, Haller P. A clustering-based approach to detect cyber attacks in process control systems. In: Proceedings of the 13th International Conference on Industrial Informatics. Cambridge, United Kingdom, 2015. 142−148

20 Yan Z, Wang J. Model predictive control of nonlinear systems with unmodeled dynamics based on feedforward and recurrent neural networks. *IEEE Transactions on Industrial Informatics*, 2012, **8**(4): 746−756

21 Feng Peng. Research and Application of Network Traffic Prediction Algorithm Based on PSO-BP Neural Network [Master thesis], Northeastern University, China, 2015.
(封鹏. 基于 PSO-BP 神经网络的网络流量预测算法的研究与应用 [硕士学位论文], 东北大学, 中国, 2015.)

22 Huang G B, Zhou H M, Ding X J, Zhang R. Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2012, **42**(2): 513−529

23 Huang G B, Zhu Q Y, Siew C K. Extreme learning machine: A new learning scheme of feedforward neural networks. In: Proceedings of the 2004 IEEE International Joint Conference on Neural Networks. Budapest, Hungary: IEEE, 2004. 985−990

24 Huang G B, Siew C K. Extreme learning machine with randomly assigned RBF kernels. *International Journal of Information Technology*, 2005, **11**(1): 16−24

25 Minh H Q, Niyogi P, Yao Y. Mercer′s theorem, feature maps, and smoothing. In: Proceedings of the 2006 International Conference on Computational Learning Theory. Berlin, Heidelberg, Germary: Springer, 2006. 154−168

26 Pan W T. A new fruit fly optimization algorithm: Taking the financial distress model as an example. *Knowledge-Based Systems*, 2012, **26**: 69−74

27 Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth, Australia, 1995. 1942−1948

28 Wei Li-Xin, Zhao Mo-Lin, Fan Rui, Zhou Hong-Xing. Parameter tuning of active disturbance rejection control based on ameliorated shark smell optimization algorithm. *Control and Decision*, 2019, **34**(4): 816−820
(魏立新, 赵默林, 范锐, 周红星. 基于改进鲨鱼优化算法的自抗扰控制参数整定. 控制与决策, 2019, **34**(4): 816−820)

29 Muller S D, Marchetto J, Airaghi S, Kournoutsakos P. Optimizationbased on bacterial chemotaxis. *IEEE Transactions on Evolutionary Computation*, 2002, **6**(1): 16−29

30 Guliyev N, Ismailov V. On the approximation by single hidden layer feedforward neural networks with fixed weights. *Neural Networks*, 2018, **98**: 296−304

31 Dai Li. Distributed Stochastic Model Predictive Control [Ph.D. dissertation], Beijing Institute of Technology, China, 2016.
(戴荔. 分布式随机模型预测控制方法研究 [博士学位论文], 北京理工大学, 中国, 2016.)

32 Liu K, Ma A Y, Xia Y Q, Sun Z Q, Johansson K H. Network scheduling and control co-design for multi-loop MPC. *IEEE Transactions on Automatic Control*, 2019, **64**(12): 5238−5245

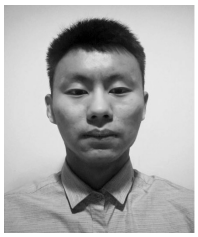
33 Marruedo D L, Alamo T, Camacho E F. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: Proceedings of the 41st IEEE Conference on Decision and Control. Las Vegas, Nevada, USA, 2002. 4619−4624

34 Xia Yuan-Qing. Cloud control systems and their challenges. *Acta Automatica Sinica*, 2016, **42**(1): 1−12
(夏元清. 云控制系统及其面临的挑战. 自动化学报, 2016, **42**(1): 1−12)

刘 坤 北京理工大学自动化学院教授. 主要研究方向为网络化控制理论与应用, 复杂网络控制与安全. 本文通信作者.
E-mail: kunliubit@bit.edu.cn
(**LIU Kun** Professor at the School of Automation, Beijing Institute of Technology. His research interest covers theory and applications of networked control, and control and security of complex networked systems. Corresponding author of this paper.)

马书鹤 北京理工大学自动化学院硕士研究生. 主要研究方向为攻击检测, 安全控制, 机器学习.
E-mail: mashuhehe@163.com
(**MA Shu-He** Master student at the School of Automation, Beijing Institute of Technology. Her research interest covers attack detection, secure control, and machine learning.)

马奥运 北京理工大学自动化学院博士研究生. 主要研究方向为模型预测控制, 优化控制.
E-mail: maaoyun92@gmail.com
(**MA Ao-Yun** Ph.D. candidate at the School of Automation, Beijing Institute of Technology. His research interest covers model predictive control and optimal control.)



张淇瑞 北京理工大学自动化学院博士研究生. 主要研究方向为信息物理系统的安全控制, 最优化控制.
E-mail: qiruizhang@bit.edu.cn
(**ZHANG Qi-Rui** Ph.D. candidate at the School of Automation, Beijing Institute of Technology. His research interest covers secure control of cyber-physical systems and optimal control.)

夏元清 北京理工大学自动化学院教授. 主要研究方向为云控制, 云数据中心优化调度管理, 智能交通, 模型预测控制, 自抗扰控制, 飞行器控制和空天地一体化网络协同控制.
E-mail: xia_yuanqing@bit.edu.cn
(**XIA Yuan-Qing** Professor at the School of Automation, Beijing Institute of Technology. His research interest covers cloud control, cloud data center optimization scheduling and management, intelligent transportation, model predictive control, active disturbance rejection control, flight control, and networked cooperative control for integration of space, air and earth.)