

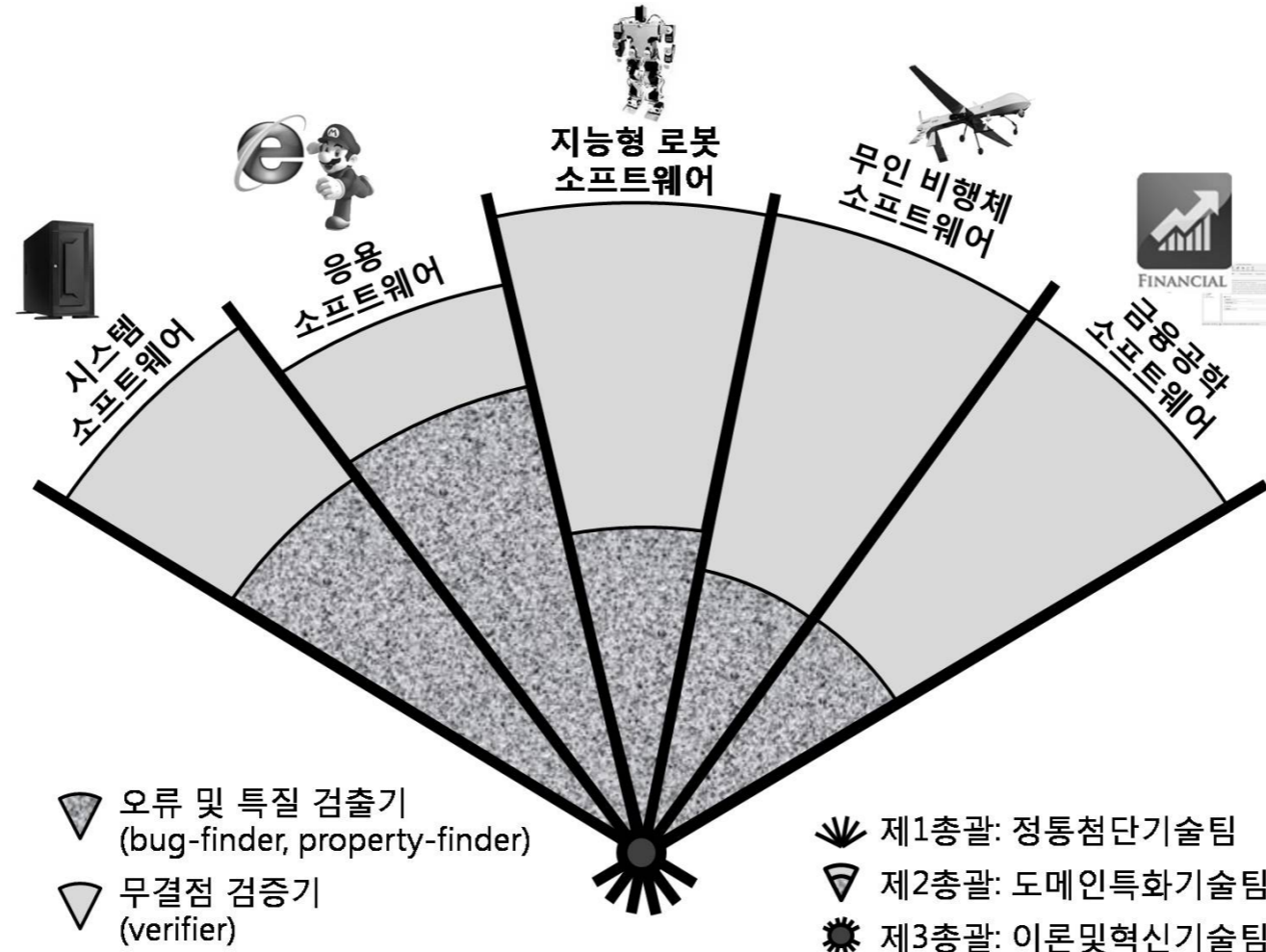
지능형 로봇 SW 분석*

서울대학교
프로그래밍 연구실
허기홍

* 프로그래밍 연구실 정영범
로봇자동화 연구실 한재영, 고인영

문제

목표



- 지능형 로봇 SW 분석
- 도메인 특화 기술로 정확도 향상

대상 프로그램

- Motion Planner
 - 로봇이 할 일을 입력받아 동작을 결정
- C++, 약 27000 줄

민감도 분석

- K 의 변화에 따른 V 값의 변화량
- K, V : 프로그램 변수
- K : 비례 상수 (실험적으로 결정)
- V : 거리, 속도, 각도 등

예

=가정=

$9 \leq V \leq 10$ 이어야 함

```
class erController{  
    ...  
    double K;  
    ...  
};
```

```
void erController::InitParameters(){  
    ...  
    K = 1.0;  
    ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
    ...  
    V = Metric(model.GetEffector(), x_d);  
    ...  
}
```

예

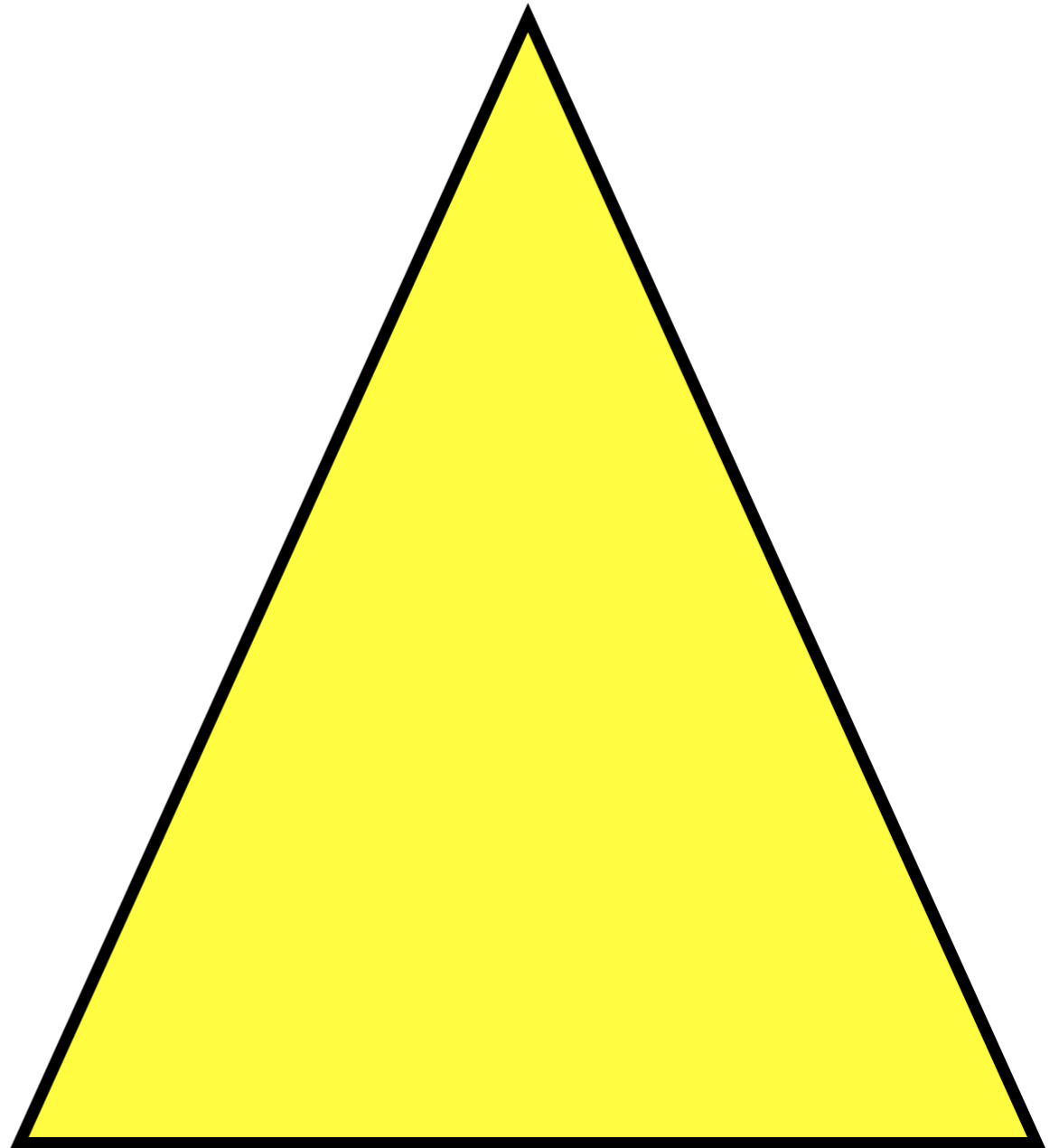
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

로봇 SW



예

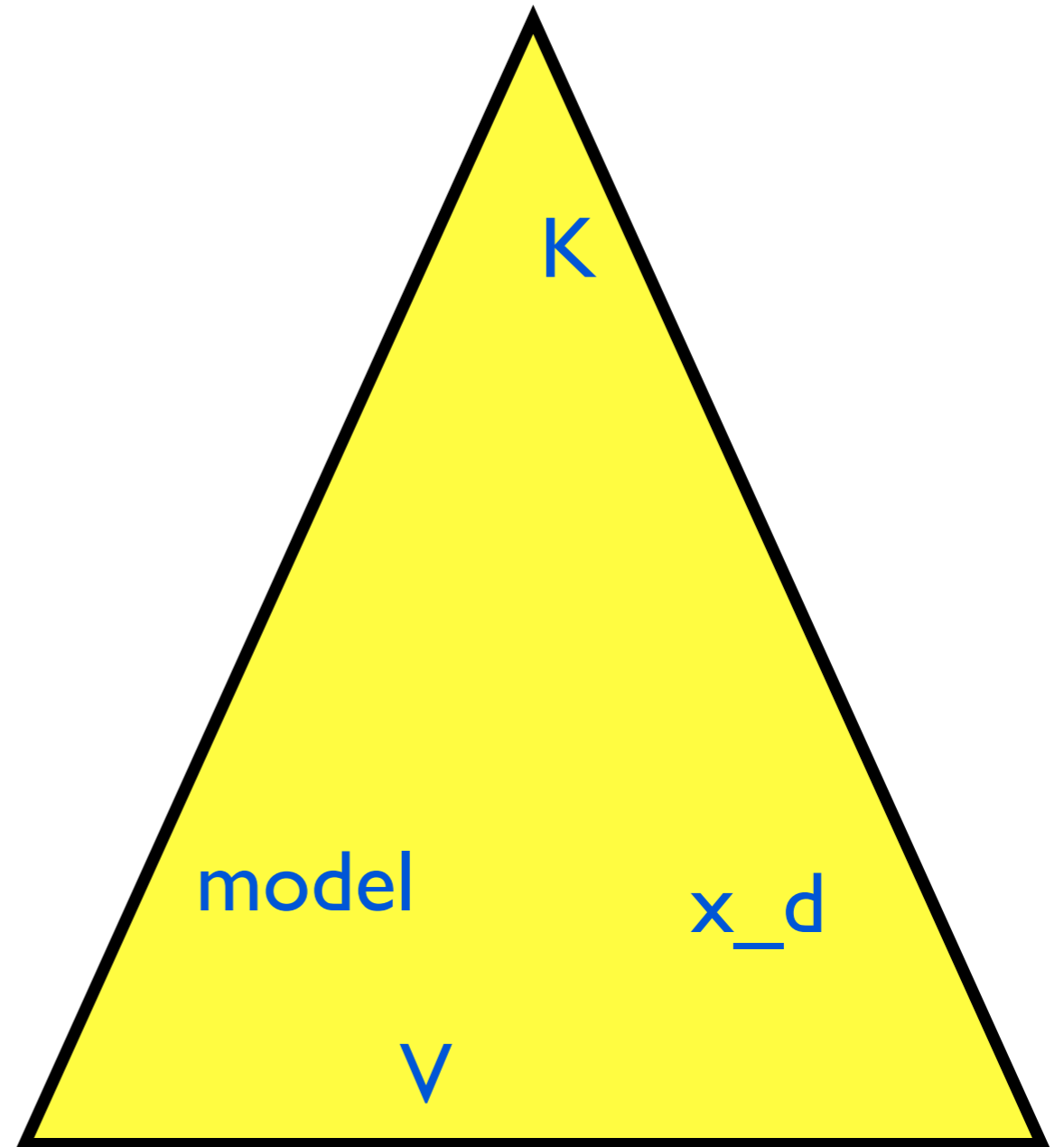
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

로봇 SW



예

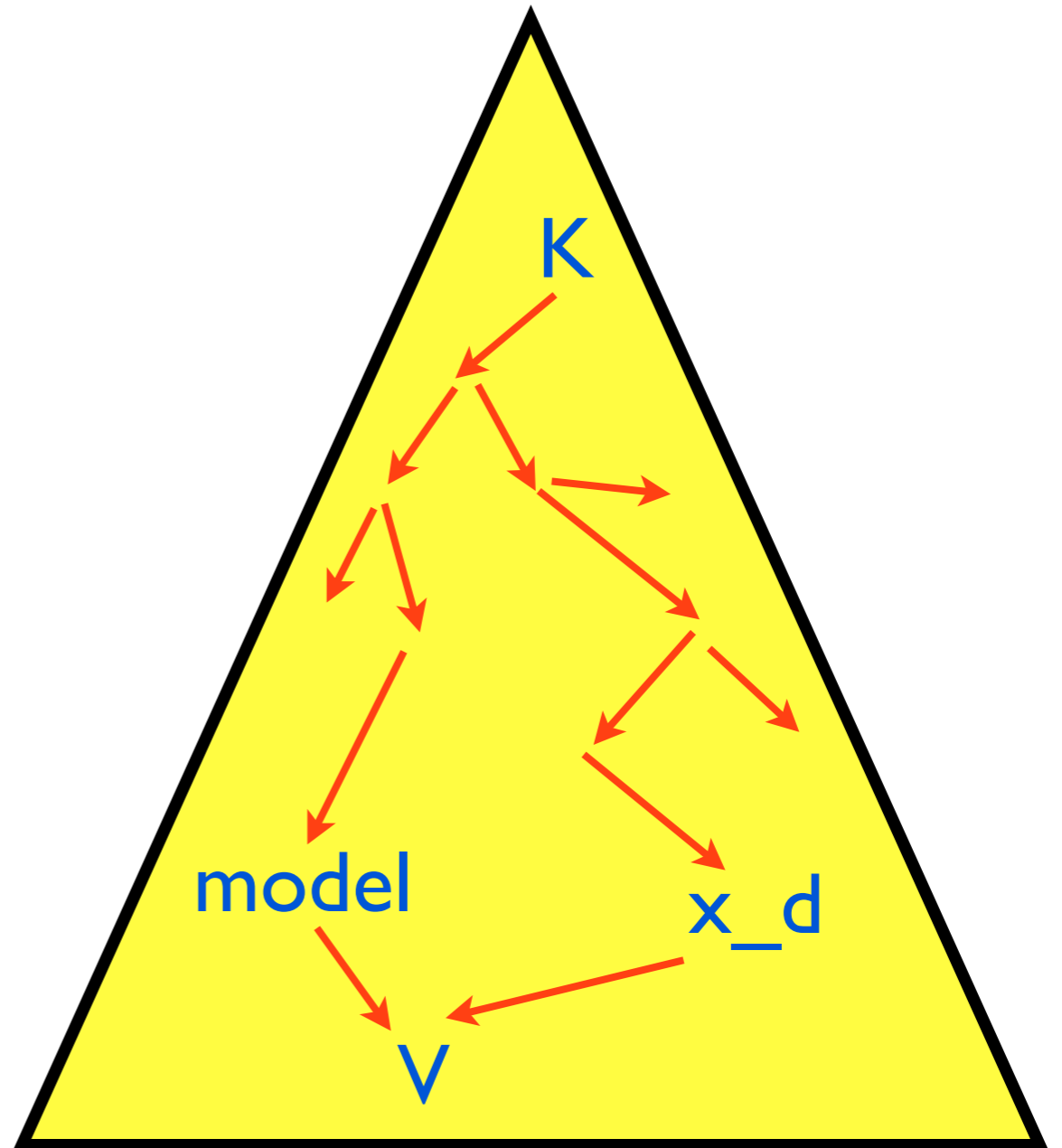
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

로봇 SW



예

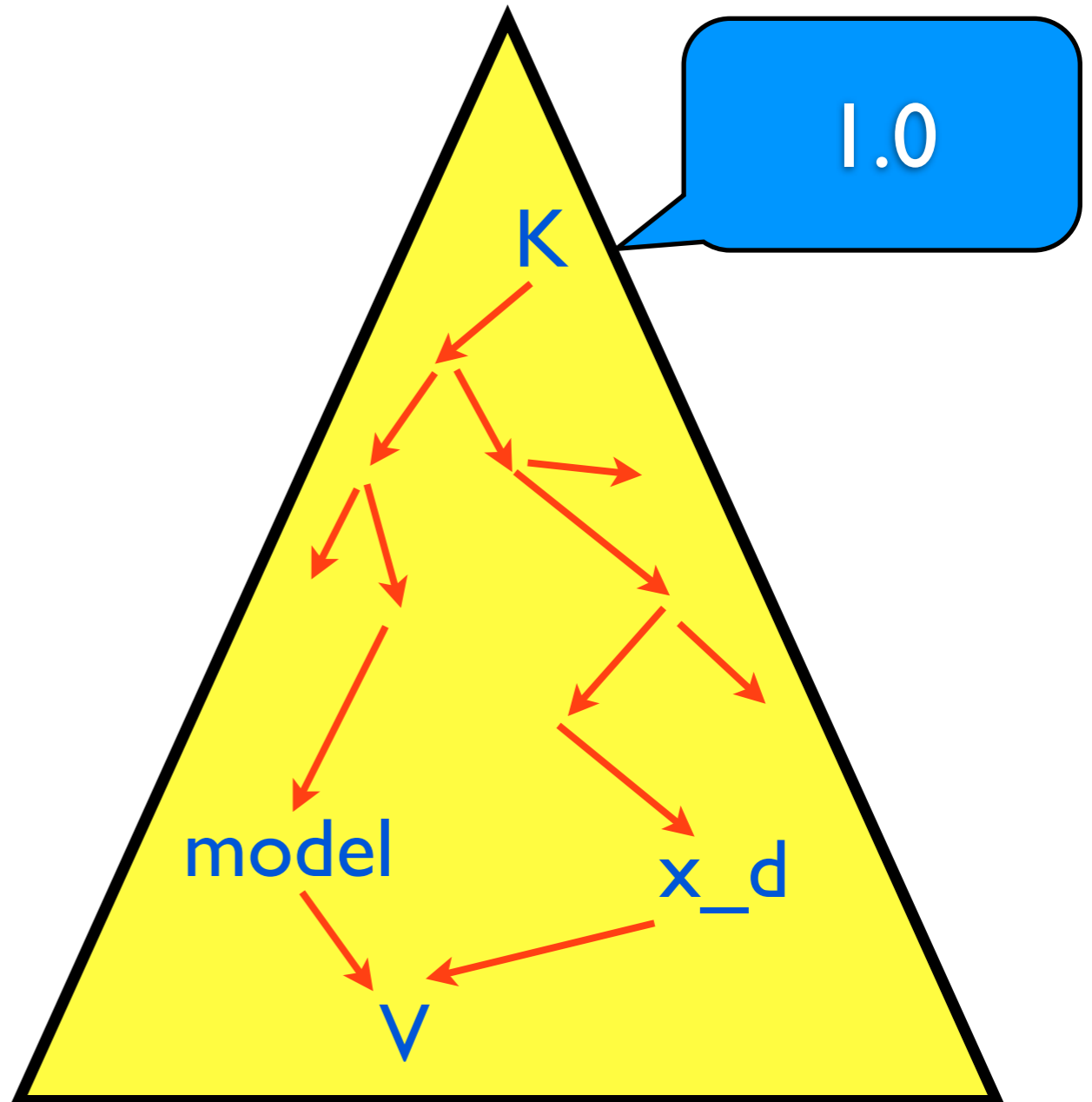
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

로봇 SW



예

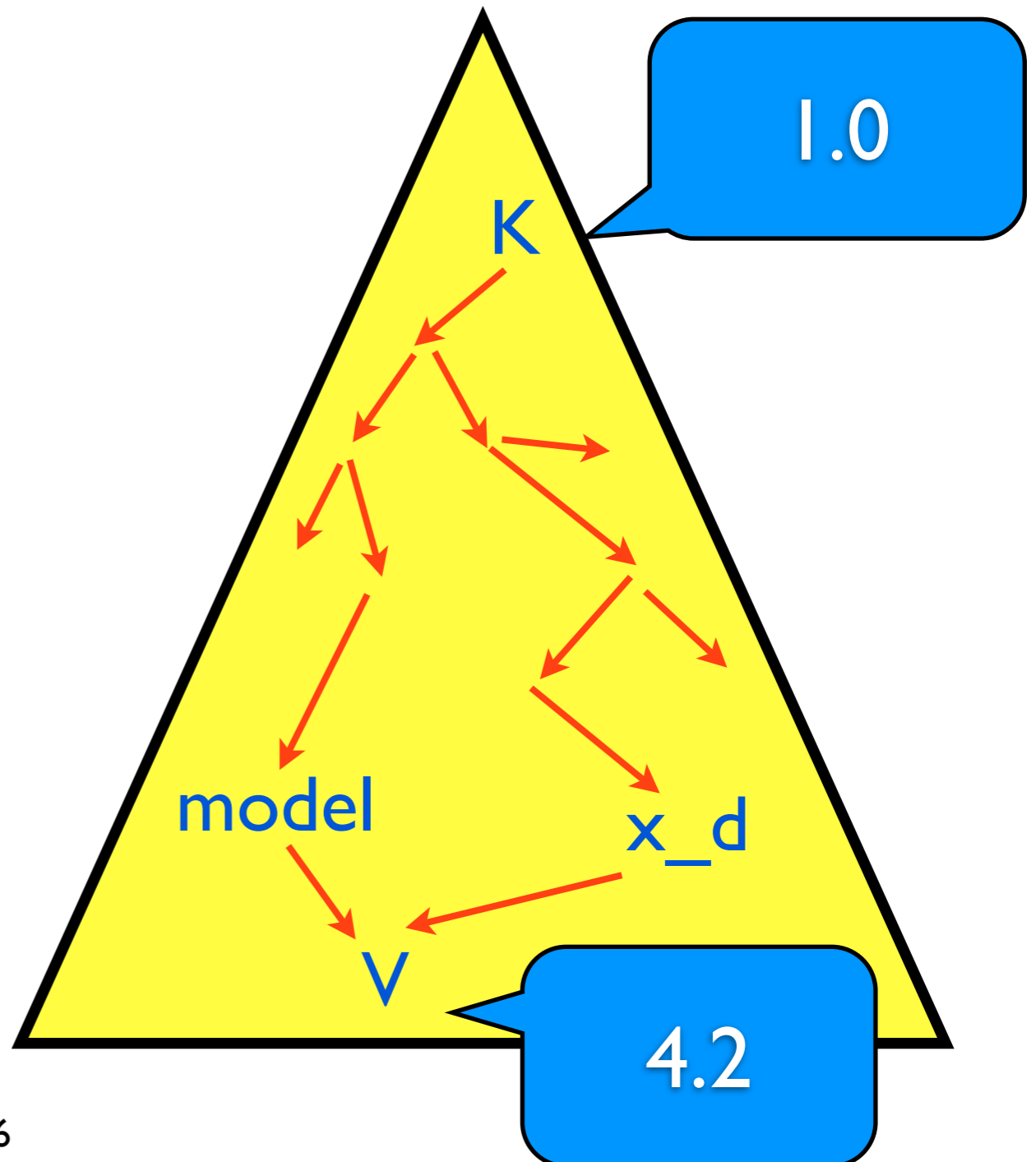
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

로봇 SW



예

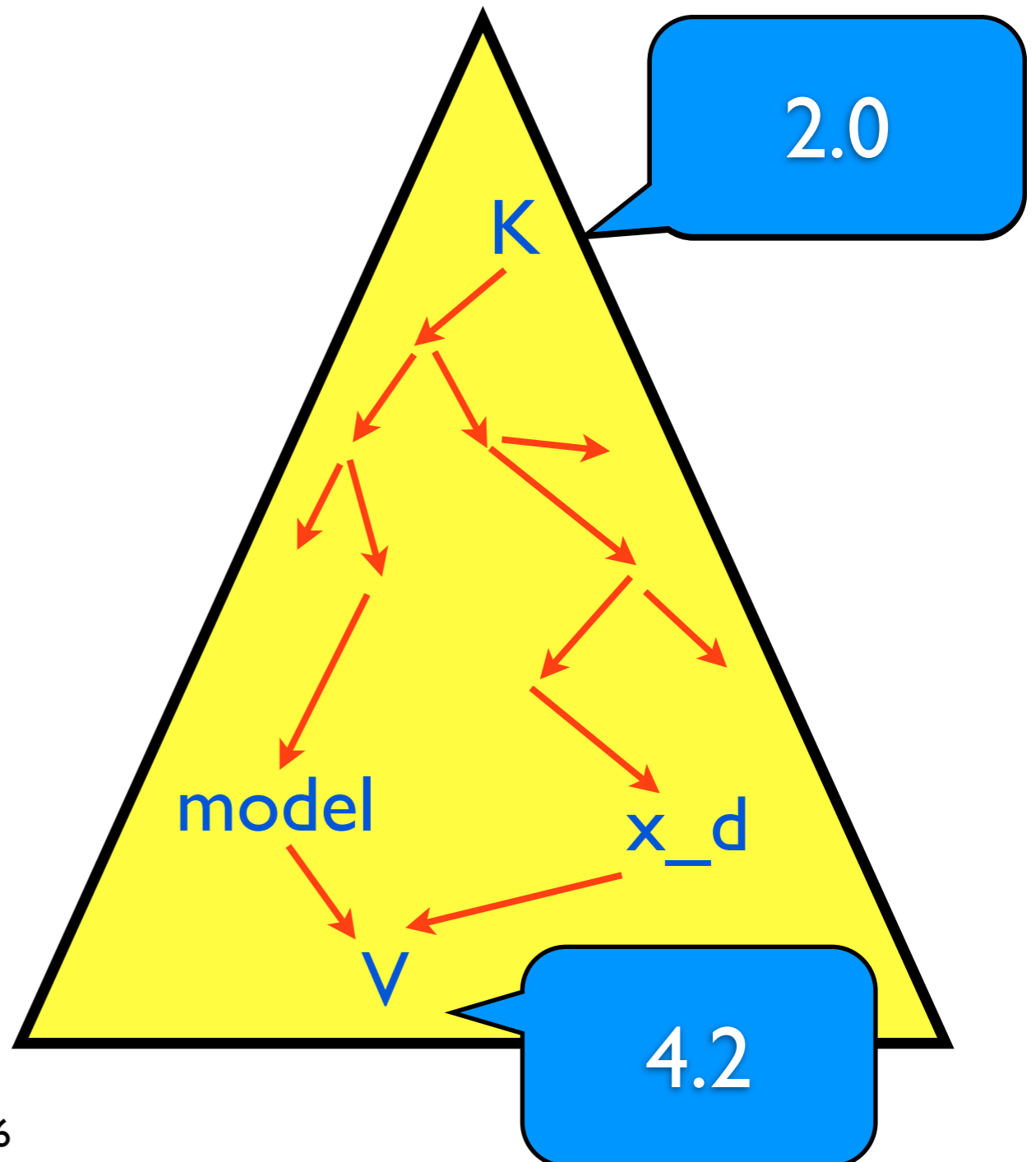
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

로봇 SW



예

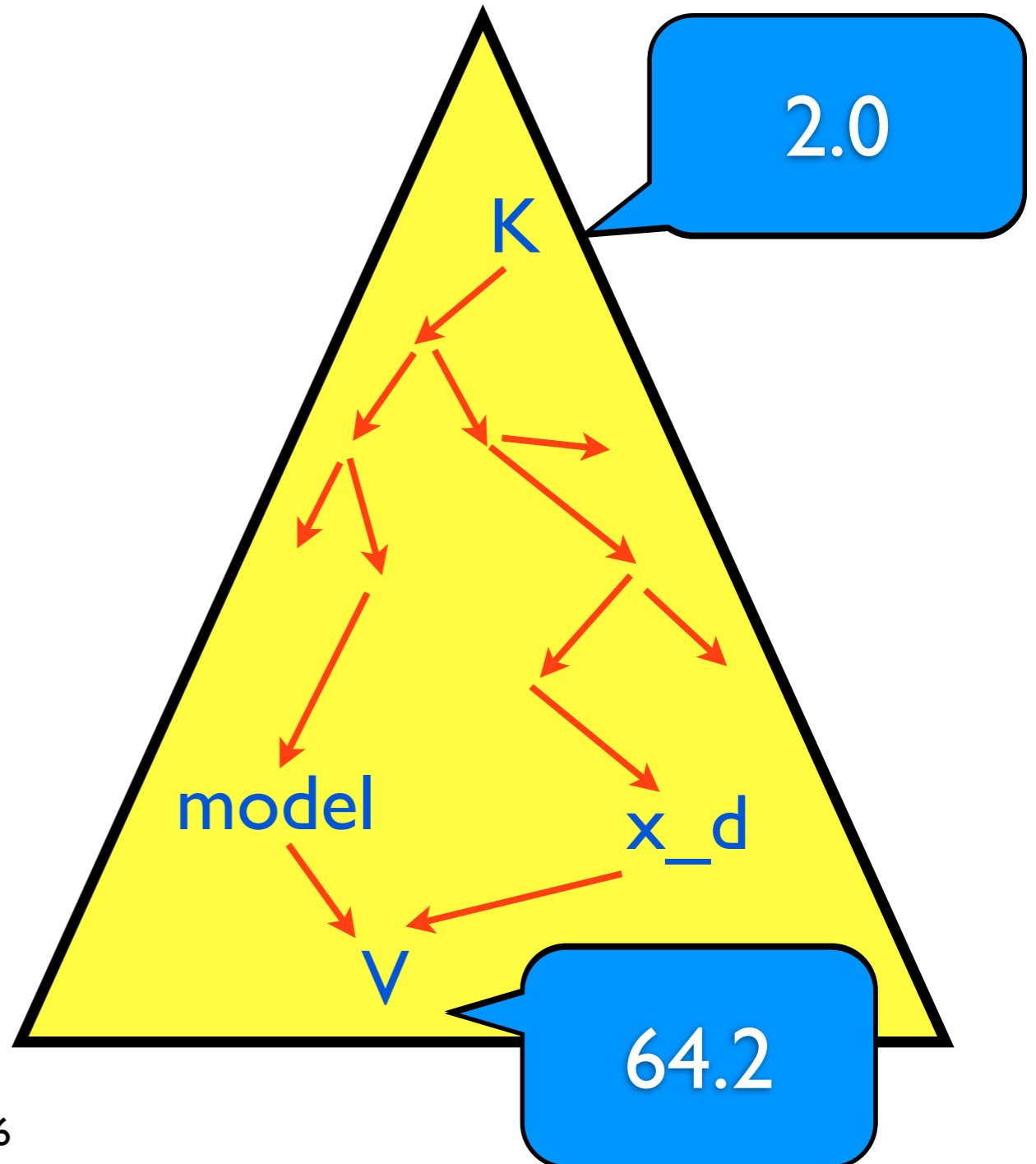
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

로봇 SW



예

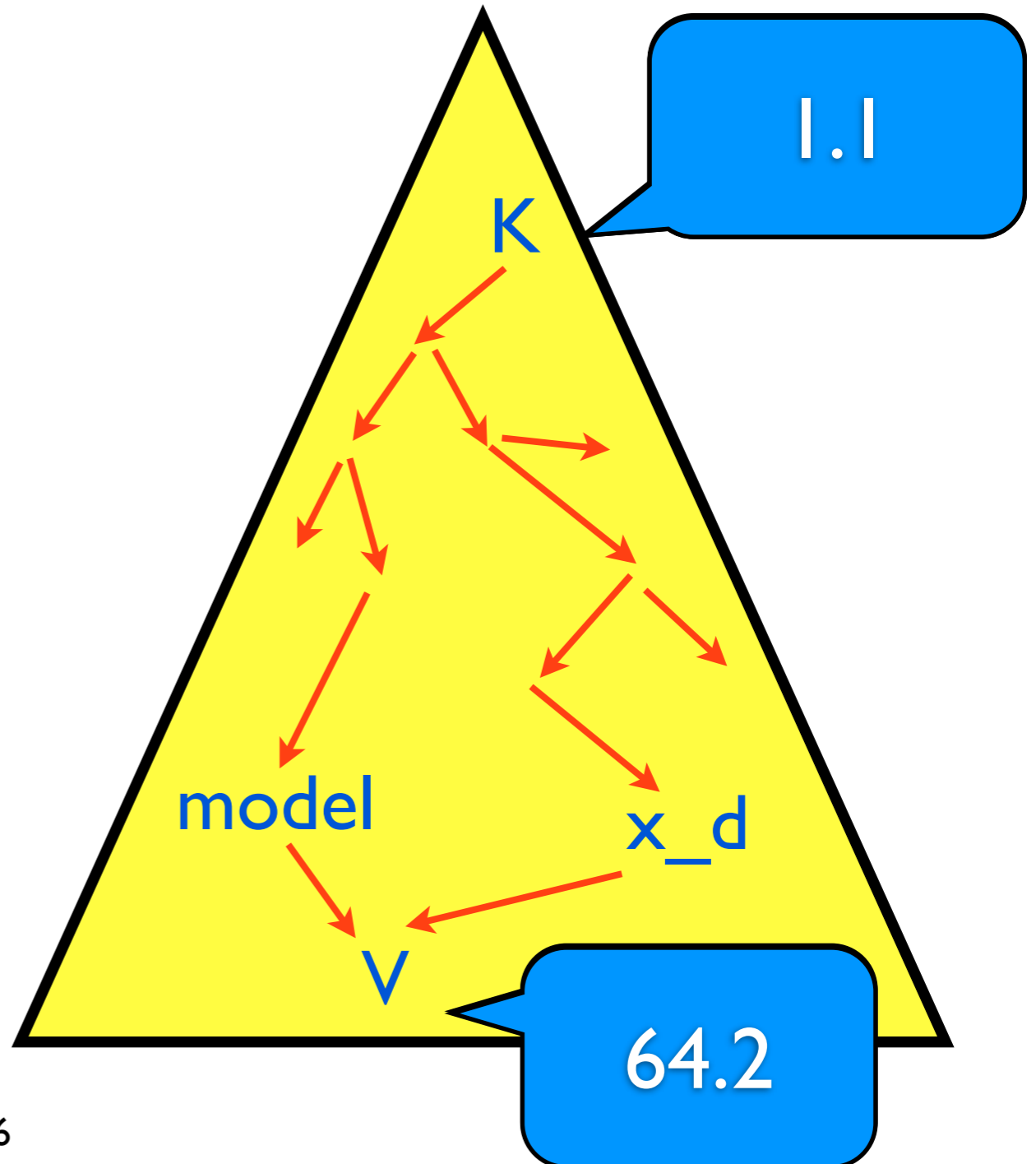
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

로봇 SW



예

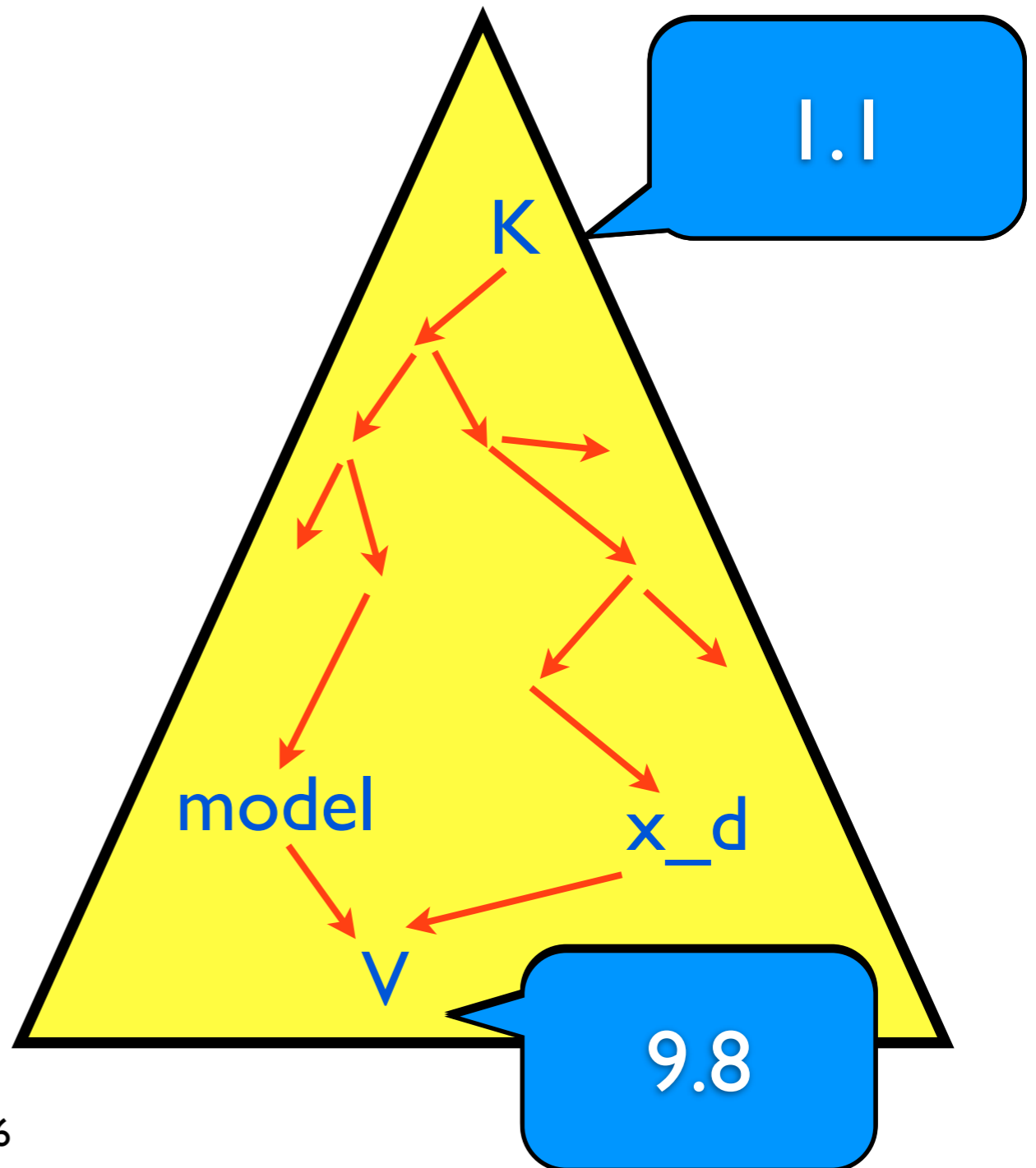
=가정=
 $9 \leq V \leq 10$ 이어야 함

```
class erController{  
  ...  
  double K;  
  ...  
};
```

```
void erController::InitParameters(){  
  ...  
  K = 1.0;  
  ...  
}
```

```
Velocity Planner::PathGenerator(int idx){  
  ...  
  V = Metric(model.GetEffector(), x_d);  
  ...  
}
```

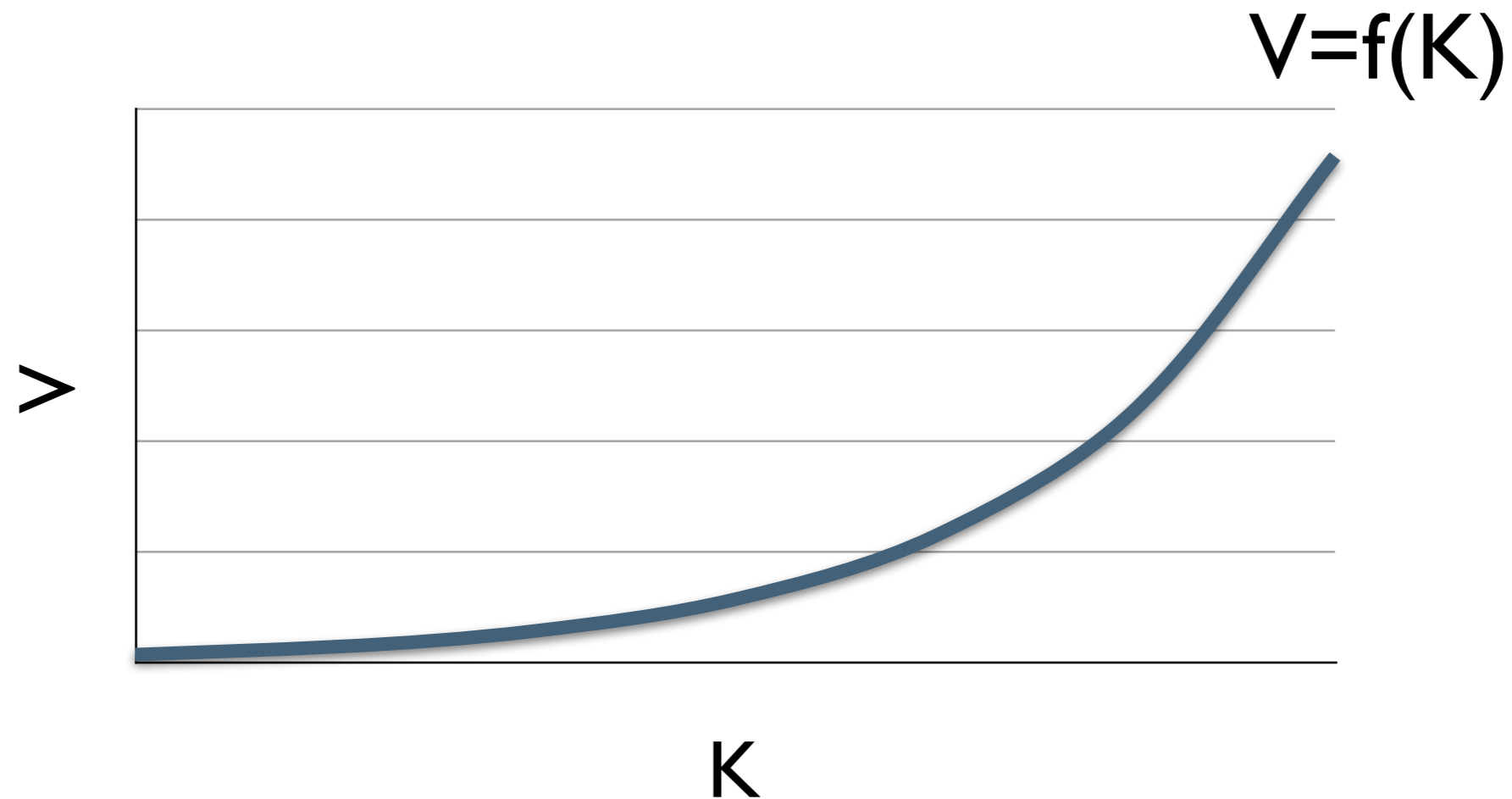
로봇 SW



필요성

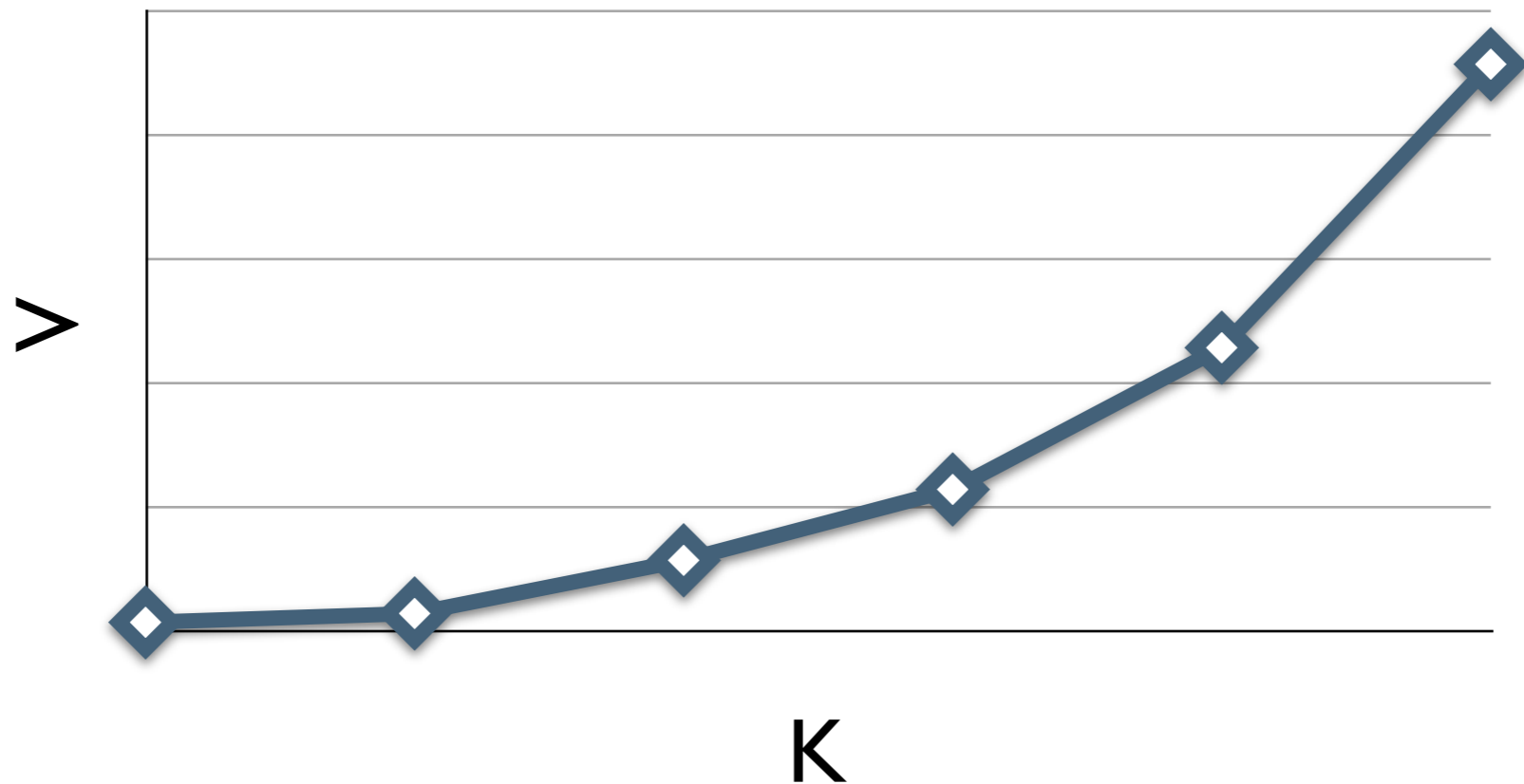
- 적당한 K값으로 튜닝하는데 유용
- K는 테스트를 통해 결정
- 경험과 직관을 보충

완_어벽_려한_은 해결책



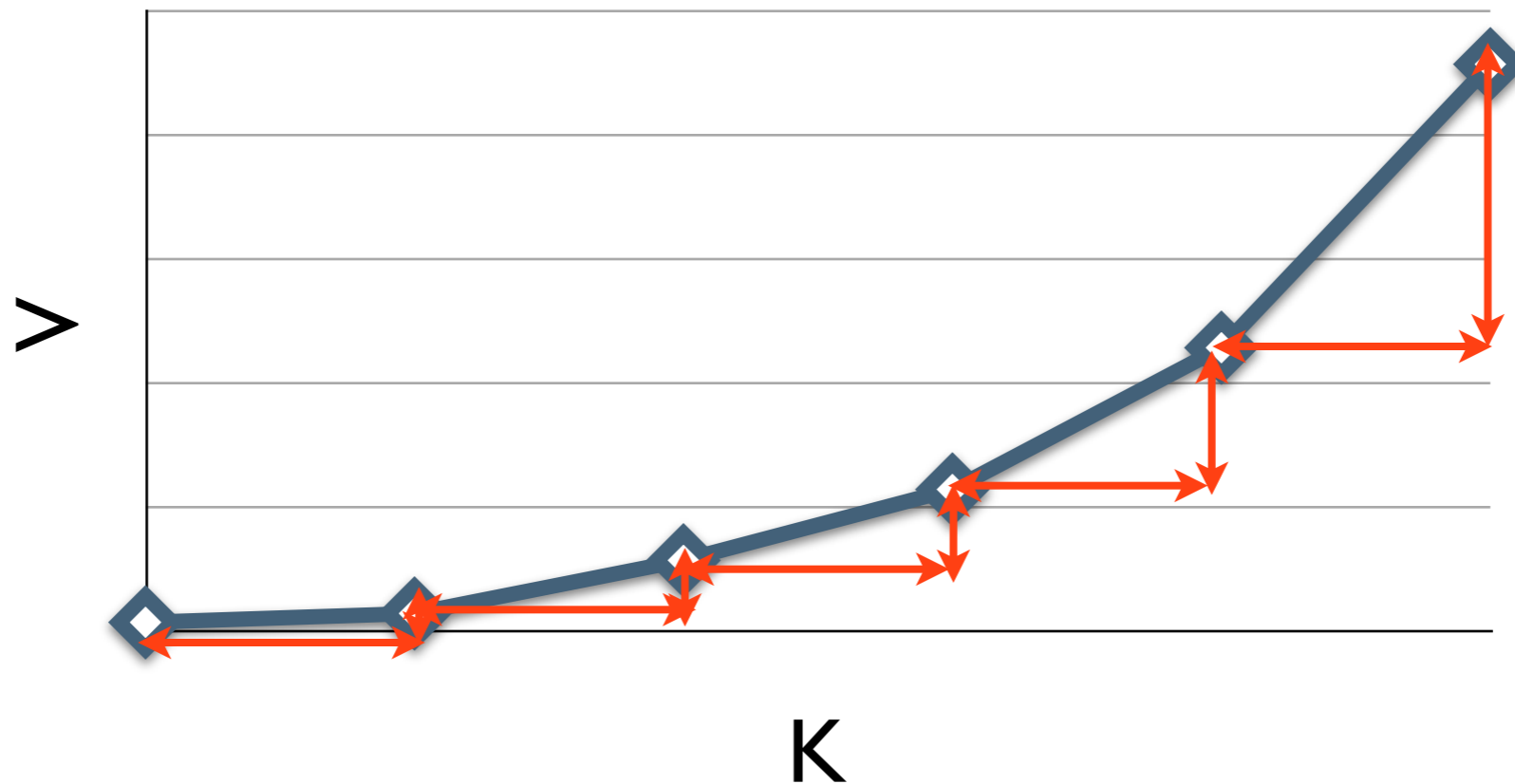
- $V=f(K)$ 함수 구하기
- 코드에서 수식을 도출하기란 어려움

요_좁약_쉬된_운 해결책



- 여러 직선을 조합하여 $f(K)$ 를 근사
- 구간 도메인을 이용

요_좁약_쉬된_운 해결책



- 여러 직선을 조합하여 $f(K)$ 를 근사
- 구간 도메인을 이용

분석

- (입력, 출력)
- (여러 구간으로 표현한 K,
여러 구간으로 표현한 V)
- 예
 - $K = \{[0.0, 1.0], [5.0, 5.5], [5.0, 5.1]\}$
 $\Rightarrow V = \{[3.6, 7.5], [34.5, 50.1], [34.5, 38.2]\}$

진행상황

디딤돌

- ‘아이락’에 기반
 - 일반적인 C 프로그램 대상
 - 정수 구간 도메인 사용
 - 버퍼 오버런 오류 분석

C++ → C

- C++ 코드를 C로 변환하여 분석
 - 심오한 OO 특징 없으므로
 - 아이락에 기댈 수 있으므로
- 변환
 - EDG 파서 : 포장용
 - 손 : 실험용

도메인 확장

- 정수 구간 \rightarrow 부동 소수점 구간
- 코드내 대부분 값은 부동 소수점

정확도 향상 기술

- 배열의 각 원소를 구분
 - 대부분 배열 크기 < 12
- 순환문 전개
 - 대부분 간단한 for 문

향후 과제 & 논의사항

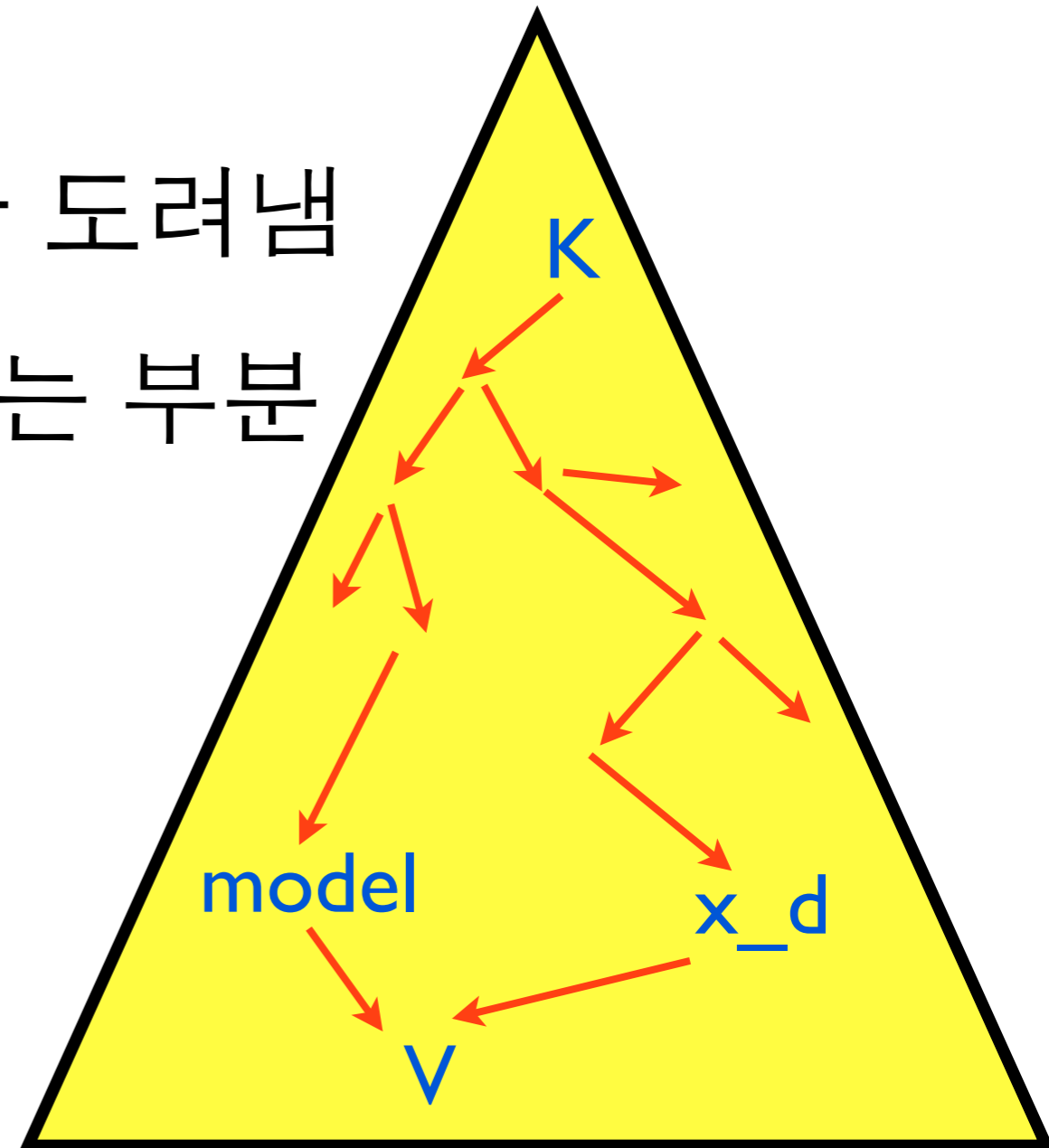
코드 슬라이싱

- 소스 중 **우리 관심사**만 도려냄
 - K가 변할 때 영향 받는 부분
- 속도 향상 가능
- 정확도 향상 가능

코드 슬라이싱

로봇 SW

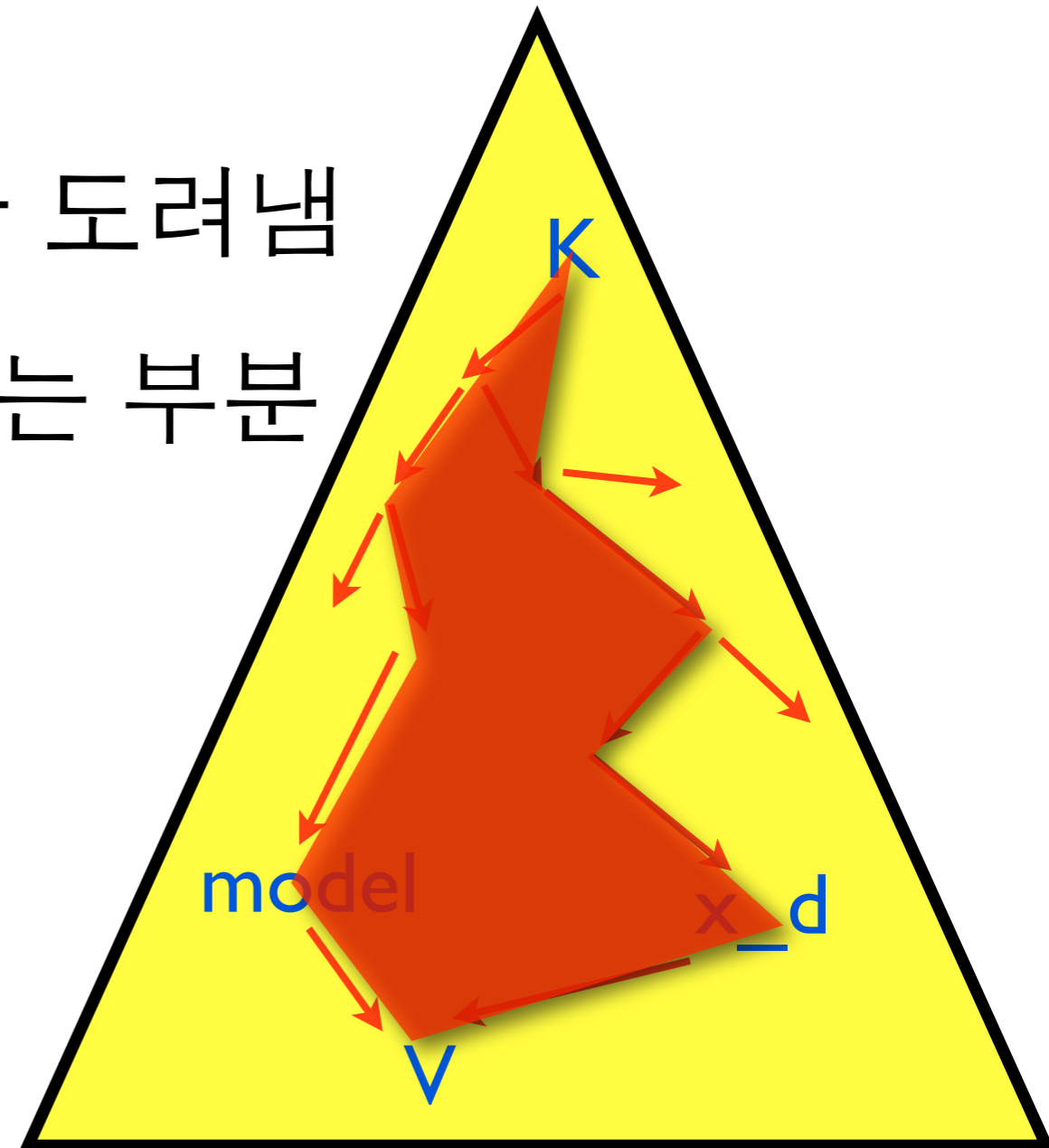
- 소스 중 **우리 관심사**만 도려냄
- K가 변할 때 영향 받는 부분
- 속도 향상 가능
- 정확도 향상 가능



코드 슬라이싱

로봇 SW

- 소스 중 **우리 관심사**만 도려냄
- K가 변할 때 영향 받는 부분
- 속도 향상 가능
- 정확도 향상 가능



속도 향상

- 자른 코드만 반복 분석
- 전체 코드 분석 : 15분 가량

정확도 향상

- 자른 부분의 특징에만 집중
 - 배열 크기 고정/변동
 - 순환문의 복잡도

고맙습니다